Northeastern University
College of Computer and Information Science

PhD Thesis Approval

THESIS TITLE: **On the Privacy Implications of Real Time Bidding**

AUTHOR: **Muhammad Ahmad Bashir**

Ph.D. Thesis Approved to complete all degree requirements for the Ph.D. Degree in Computer Science.

Christo Wilson — 8/13/19
Thesis Advisor — *Date*

David Choffnes — 8/13/19
Thesis Reader — *Date*

William Robertson — 8/13/19
Thesis Reader — *Date*

Arvind Narayanan Arvind. N. — 08/19/2019
Thesis Reader — *Date*

Thesis Reader — *Date*

GRADUATE SCHOOL APPROVAL:

8/21/19
Director, Graduate School — *Date*

COPY RECEIVED IN GRADUATE SCHOOL OFFICE:

SARAH GALE — 8/19/19
Recipient's Signature — *Date*

Distribution: Once completed, this form should be scanned and attached to the front of the electronic dissertation document (page 1). An electronic version of the document can then be uploaded to the Northeastern University-UMI website.

On the Privacy Implications of Real Time Bidding

A Dissertation Presented
by

Muhammad Ahmad Bashir

to

The Khoury College of Computer Sciences

in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Computer Science

Northeastern University
Boston, Massachusetts

August 2019

To my parents, Javed Hanif and Najia Javed
for their unconditional support, love, and prayers.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my advisor, Christo Wilson. None of this work would have been possible without his support and guidance. I am extremely grateful for the faith he showed in me all these years. He has always been my advocate and my champion. He provided resources and removed hurdles for me so that I can achieve my goals. I always enjoyed and looked forward to our conversations during the meetings, whether they were about a research project or a new Netflix show. They say finding the right advisor can be hard. Well, I guess I got lucky because I could not have asked for a better one.

I would also like to thank David Choffnes, William Robertson, and Arvind Narayanan for serving on my committee. Their insightful advice has been very helpful in this thesis.

I would like to acknowledge all my collaborators, from whom I have learned a lot: William Robertson, Engin Kirda, Alan Mislove, Fareed Zaffar, Umar Farooq, Maryam Shahid, Basit Shafiq, Bilal Zafar, Bimal Viswanath, Krishna Gummadi, and Saikat Guha. Special thanks to Sajjad Arshad for working on the *Inclusion Crawler* tool, which has been instrumental in my research.

I am very thankful to Krishna Gummadi and Bimal Viswanath for mentoring me during my time at MPI-SWS. I am extremely grateful to Alan and Christo for bringing me to Northeastern when I faced visa issues from Germany. I would also like to thank the mentors I have had during my internships: Eleni and Nektarios at Facebook, and Narseo at ICSI (Berkeley).

I would also like to thank the many wonderful friends I made in Boston: Konstantinos, Le, Arash, Hamid, Lulu, Chin, Apoorve, Aditeya, Mishuk, Michael, Andreas, Chaima, Andrea, Sammie, Marinos, Lydia, Fangfan, Jingjing, Giri, Tijay, Piotr, Giorgos, Ali, Talha, and Amogh. They all made living in Boston a lot more fun.

I am extremely fortunate to have so many friends from back home in the US. I am grateful for Haris, Anum, Raza, Saman, Mehr, Nayab, Anique, and Nimra, who have been my family away from home.

Finally, and most importantly, I would like to express my deep gratitude to my family, especially my parents for their unconditional support during all these years. Keeping in touch with my siblings always brought me joy and made my life away from home much easier. Special thanks to my partner Amber, for always believing in me and putting up with my grad school lifestyle.

Abstract of the Dissertation

On the Privacy Implications of Real Time Bidding

by

Muhammad Ahmad Bashir

Doctor of Philosophy in Computer Science

Northeastern University, August 2019

Dr. Christo Wilson, Advisor

The massive growth of online advertising has created a need for commensurate amounts of user tracking. Advertising companies track online users extensively to serve targeted advertisements. On the surface, this seems like a simple process: a tracker places a unique cookie in the user's browser, repeatedly observes the same cookie as the user surfs the web, and finally uses the accrued data to select targeted ads.

However, the reality is much more complex. The rise of Real Time Bidding (RTB) has forced the Advertising and Analytics (*A&A*) companies to collaborate more closely with one another, to exchange data about users to facilitate bidding in RTB auctions. The amount of information-sharing is further exacerbated by how real-time auctions are implemented. During an auction, several A&A companies observe user *impressions* as they receive bid requests, even though only one of them eventually wins the auction and serves the advertisement. This significantly increases the privacy *digital footprint* of the user. Because of RTB, tracking data is not just observed by trackers embedded directly into web pages, but rather it is funneled through the advertising ecosystem through complex networks of exchanges and auctions.

Numerous surveys have shown that web users are not completely aware of the amount of data sharing that occurs between A&A companies, and thus underestimate the privacy risks associated with online tracking. To accurately quantify users' privacy digital footprint, we need to take into account the information-sharing that happens either to facilitate RTB auctions or as a consequence of them.

However, measuring these flows of tracking information is challenging. Although there is prior work on detecting information-sharing (*cookie matching*) between A&A companies, these studies are based on brittle heuristics that cannot detect all forms of information-sharing (e.g., server-side matching), especially under adversarial conditions (e.g., obfuscation). This limits our view of the privacy landscape and hinders the development of effective privacy tools.

The overall goal of my thesis is to understand the privacy implications of Real Time Bidding, to bridge the divide between the *actual* privacy landscape and our understanding of it. To that end, I propose methods and tools to accurately map information-sharing among A&A domains in the modern ad ecosystem under RTB.

First, I propose a content-agnostic methodology that can detect client- and server-side information flows between arbitrary A&A domains using *retargeted ads*. Intuitively, this methodology works because it relies on the *semantics* of how exchanges serve ads, rather than focusing on specific cookie matching *mechanisms*. Using crawled data on 35,448 ad impressions, I show that this methodology can successfully categorize four different kinds of information-sharing behaviors between A&A domains, including cases where existing heuristic methods fail.

Next, in order to capture the effects of ad exchanges during RTB auctions accurately, I isolate a list of A&A domains that act as ad exchanges during the bidding process. Identifying such A&A domains is crucial, since they can disperse user impressions to multiple other A&A domains to solicit bids. I achieve this by conducting a longitudinal analysis of a transparency standard called `ads.txt`, which was introduced to combat ad fraud by helping ad buyers verify authorized digital ad sellers. In particular, I conduct a 15-months longitudinal study of the standard to gather a list of A&A domains that are labeled as ad exchanges (authorized sellers) by publishers in their `ads.txt` files. Through my analysis on Alexa Top-100K, I observed that over 60% of the publishers who run RTB ads have adopted the `ads.txt` standard. This widespread adoption allowed me to explicitly identify over 1,000 A&A domains belonging to ad exchanges.

Finally, I use the list of ad exchanges from `ads.txt` along with the information flows between A&A companies collected using my generic methodology to build an accurate model of the privacy digital footprint of web users. In particular, I use these data sources to model the advertising ecosystem in the form of a graph called an *Inclusion* graph. Through simulations on the *Inclusion* graph, I provide upper and lower estimates on the tracking information observed by A&A companies. I show that the top 10% A&A domains observe at least 91% of an average user's browsing history under reasonable assumptions about information-sharing within RTB auctions. I also evaluate the effectiveness of blocking strategies (e.g., AdBlock Plus) and find that major A&A domains still observe 40–90% of user impressions, depending on the blocking strategy.

Overall, in this dissertation, I propose new methodologies to understand the privacy implications of Real Time Bidding. The proposed methods can be used to shed light on the opaque ecosystem of programmatic advertising and enable users to gain a more accurate view of their digital footprint. Furthermore, the results of this thesis can be used to build better or enhance existing privacy-preserving tools.

Chapter 1

Introduction

In the last decade, the online display advertising industry has massively grown in size and scope. In 2017, $83 billion were spent on digital advertising in the U.S., and double-digit growth is forecast for each subsequent year, surpassing $125 billion in digital advertising expenditure by 2022 [3,160]. This increased spending is fueled by advances in the industry that enable ad networks to track and target users with increasing levels of precision.

People have complicated feelings with respect to online behavioral advertising. While surveys have shown that some users prefer relevant, targeted ads to random, untargeted ads [36, 188], this preference has caveats. For example, users are uncomfortable with ads that are targeted based on sensitive Personally Identifiable Information (PII) [11, 120] or specific kinds of browsing history (e.g., visiting medical websites) [115]. Furthermore, some users are universally opposed to online tracking, regardless of circumstance [36, 124, 188].

One particular concern held by users is their *digital footprint* [92, 180, 197], which I define as the first- and third-parties that are able to track their browsing history (see § 2.1). Large-scale web crawls have repeatedly shown that trackers are ubiquitous [61, 62, 74], with DoubleClick alone being able to observe visitors on 40% of websites in the Alexa Top-100K [33]. These results paint a picture of a balkanized web, where trackers divide up space and compete for the ability to collect data and serve targeted ads. On the surface, this seems like a simple process: a tracker places a unique cookie (identifier) in the user's browser, repeatedly observes the same cookie as the user surfs the web, and finally uses the accrued data to display them targeted ads.

However, the reality is much more complex. The modern online ad ecosystem has seen a massive shift towards an auction-based model called Real Time Bidding (RTB), where ad networks bid on individual user impressions. As of 2018, RTB holds a 30% share in the digital advertising spending in the U.S. [1] and has a forecasted annual growth rate of 33% between 2019 and 2024 [4]. Since the advent of RTB, hundreds of specialized companies with various business models have emerged in the market. On one hand, *Supply Side Platforms* (SSPs) help publishers (e.g., CNN, ESPN) maximize their revenue by helping them maintain business relationships with lucrative ad exchanges. On the flip side, *Demand Side Platforms* (DSPs) work closely with advertisers (e.g., Nike, Pepsi) to evaluate the value of each user impression and optimize bid prices during RTB auctions. *Ad exchanges* implement RTB auctions, where DSPs bid on each user impression being sold by publishers via SSPs. I provide more details on the RTB ecosystem in § 2.3. In this dissertation, I collectively refer to companies engaged in *Advertising and Analytics* as A&A companies.

While the RTB model brings more flexibility in the ad ecosystem, it has privacy implications for users. First, the rise of RTB has forced A&A companies to collaborate more closely with one another by sharing unique user identifiers through a process called *cookie matching*. Cookie matching is a pre-condition for A&A companies to participate in RTB auctions and because of this, tracking data is not just observed by A&A companies embedded directly into publishers' web pages, but rather the data is funneled through the advertising ecosystem via complex networks of SSPs, exchanges, and DSPs.

Second, during the RTB auction, ad exchanges forward user impressions to several partner DSPs to solicit bids from them. The vast majority of RTB auction are held on the server (exchange) side and not on the client (browser) side, which means that the browser only gets redirected to the winner of the auction and does not observe those DSPs that participate in the auction but did not win. Although only the winner serves the user an advertisement, *all* participating DSPs view the user impression. This further increases the user's privacy digital footprint, potentially without their knowledge.

Due to the close collaboration among A&A companies, we can no longer view them as isolated islands of data. To capture a more realistic picture of the privacy landscape, we have to take into account the information sharing among all A&A companies. While some users are aware that they are being tracked online [36, 188], they may not know how much and how often their information changes hands due to cookie matching and RTB. To date, technical limitations and incomplete data have prevented researchers from developing accurate models to demonstrate the privacy implications of RTB in the modern ad ecosystem. Due to this, we under-estimate the privacy digital footprint of users, which, in turn, affects the development of effective privacy tools.

By understanding how the modern advertising ecosystem works, while taking into account the effects of RTB, we may be able to develop better privacy tools for users. These tools can bring more transparency to the complex advertising industry and give more control to users over their privacy. For example, as shown in [11,120], some users are uncomfortable with ads that are targeted based on sensitive PII. With better privacy tools, these users would be able to control the attributes advertisers use to target them.

However, despite the pressing need to understand the complexities of the advertising ecosystem, we currently lack the tools to fully understand how information is being shared among A&A companies. Furthermore, we don't have a systematic way to enumerate all the participating DSPs in a given auction. These limitations hinder our understanding of *actual* privacy leakage, beyond the obvious third-party trackers that are directly embedded in web pages.

This dissertation posits that RTB has increased collaboration among A&A companies, which, in turn, has increased privacy exposure for end-users. We need effective tools and methodologies to understand these privacy implications of RTB for users, to bridge the divide between the actual privacy landscape and our understanding of it. These techniques can provide a more realistic view of the online advertising ecosystem, and enable users to gain a more accurate view of their privacy digital footprint.

1.1 Problem Statement

The goal of this dissertation is to study the privacy implications of RTB. In particular, I focus on understanding the information sharing among A&A companies, which happens either to facilitate the RTB auctions via cookie matching or as a consequence of them. In this section, I will concretely describe the problems with respect to mapping out these information-sharing relationships. In § 1.2, I give an overview of the contributions this thesis makes to address these issues.

1.1.1 Information Sharing Through Cookie Matching

During an RTB ad auction, the ad exchange solicits bids from participating DSPs. This auction happens on the server-side, which means that the DSPs receive bid requests directly from the ad exchange and **not** from the browser itself. This is problematic for the DSPs, since they cannot identify the user from the bid request, and without doing so, they cannot place meaningful bids during the auction. The inability of DSPs to identify the user stems from the fact that the bid request is not coming from the browser, which would have included the DSPs' cookie in the HTTP request. Furthermore, due to the Same Origin Policy [135] restrictions, the ad exchange cannot read cookies set by DSPs from the browser and share them with the DSPs through the bid request. To circumvent this issue, ad exchanges and DSPs sync identifiers beforehand, usually through a process known as cookie matching.

Although there has been prior empirical work on detecting information-sharing between A&A companies [5, 65, 147], these works have three fundamental limitations. First, they rely on heuristics that look for specific string signatures in HTTP messages to identify cookie matching. These heuristics are brittle in the face of obfuscation: for example, DoubleClick cryptographically hashes their cookies before sending them to other advertising partners [2, 147]. Hence, existing techniques will fail to capture obfuscated information flows. Furthermore, ad exchanges that do not rely on obfuscation might do so in the future to evade detection. I demonstrate in § 4.3 that heuristics from prior work can miss up to 31% of information sharing partners. To identify information flows in the face of obfuscation, we need to come up with a **content-agnostic** methodology.

Second, analysis of *client-side* HTTP messages is insufficient to detect *server-side* information flows between A&A companies. This can happen if two ad networks decide to sync up user tracking identifiers off the browser. For example, two ad networks that belong to the same parent company can share user tracking data on the server-side, i.e., without cookie matching. In fact, Google states in its privacy terms that "... we may combine the information we collect among our services and across your devices for the purposes ..." [79]. In § 4.3, I highlight Google services that share identifiers on the server-side. So, if the information is not shared through the browser, the analysis of HTTP messages will not yield the sharing partners. This is why we need a **platform-agnostic** methodology that can detect information sharing on both client- and server-side.

Finally, existing methods cannot determine the precise information flows between A&A companies, i.e., which parties are sending or receiving information [5]. The fundamental problem is that HTTP requests, and even the DOM tree itself, do not reveal the true sources of resource inclusions in the presence of dynamic code (JavaScript, Flash, *etc.*) from third-parties. For example, if a script t_1 embedded directly in `pub.com`, shares identifiers with t_2 using dynamic AJAX, we will *incorrectly* determine `pub.com` as the `Referer`, instead of t_1. This hides t_1's role as the source of the flow. In fact, in chapter 6, I show that the `Referer` value is incorrect 48% of the time due to dynamic inclusions. This misses or creates erroneous information-sharing relationships. To gain an accurate representation of information relationships, we need a methodology that provides **strong attribution** for resource inclusions.

In chapter 4, I propose a novel methodology that can detect information-sharing relationships in a content- and platform-agnostic manner, while providing strong attribution.

1.1.2 Information Sharing Through Ad Exchanges During RTB Auctions

This kind of information leakage happens *during* the auction process when the ad exchange solicits bids from its DSP partners. To maximize revenue, the exchange sends a bid request to several DSP partners. After evaluating the request, participating partners may submit bids. Although only the winner of the auction gets to serve the advertisement, all other *losing* participants also get to see the user impression. For example, an ad exchange e might contact ten DSP partners $d_1, d_2, ..., d_{10}$ to solicit bids for a user u to display advertisement on `cnn.com`. Irrespective of the auction winner, all ten DSPs will learn that u visited `cnn.com`. This significantly increases the users' digital footprint.

Without determining all the DSP participants in RTB auctions, we cannot gauge the extent of privacy leakage for the user. However, since the auction happens on the server-side, it limits our visibility into the ecosystem, and we can only observe the winner of the auction. One possible way to enumerate all DSP partners for a given exchange e is to observe the outcome of multiple RTB auctions held by e over time. This allows us to observe different auction winners over time, hopefully covering all the DSP partners of e. However, data collection and analysis for this task becomes challenging for the following two reasons:

1. We do not have a comprehensive list of ad exchanges. Identification of such a list is important since ad exchanges have this extra “power" to disperse tracking information to multiple DSPs during RTB auction. Without an *accurate* list of ad exchanges, we cannot precisely model the digital footprint of the user.
2. RTB auctions are complicated. The winner of the auction does not necessarily have to serve the advertisement. Frequently, the winner is an ad exchange itself and holds a new ad auction to re-sell the advertisement space. In other words, it is common for A&A companies to assume multiple roles in the ecosystem. For example, an ad network that provides DSP services might act as an ad exchange in certain circumstances. For this reason, we need to account for multiple, iterative RTB auctions when modeling users' digital footprint.

1.2 Contributions

Next, I give an overview of the solutions I propose to address the problems discussed in § 1.1. Our goal is to capture the effects of RTB auctions to get a better picture of the privacy digital footprint of the user. To this end, this thesis makes the following contributions.

1.2.1 A Generic Methodology For Detecting Information Sharing Among A&A companies

Given the limitations of existing techniques [5,65,147] I describe in § 1.1.1, the first contribution I make in this thesis is proposing a novel methodology that can detect client- and server-side flows of information between arbitrary A&A companies using *retargeted ads*. Retargeted ads are the most specific form of behavioral advertisements, where a user is targeted with ads related to the exact products she has previously browsed (see § 2.3 for definition). For example, Bob visits nike.com and browses for running shoes but decides not to purchase them. Bob later visits cnn.com and sees an ad for the exact same running shoes from Nike.

My key insight is to leverage retargeted ads as a mechanism for identifying information flows between arbitrary A&A companies. This is possible because the strict conditions that must be met for a retargeted ad to be served, allow us to infer the precise flow of tracking information that facilitated the serving of the ad. Intuitively, this methodology works because it relies on the *semantics* of how exchanges serve ads, rather than focusing on specific cookie matching *mechanisms*. Specifically, instead of relying on HTTP messages to detect cookie matching, it relies on causality; i.e., if ad network a_1 observes user u browsing a product p on shop s, and if later a_1 serves u a retargeted ad for p after winning the RTB auction held by ad exchange e_1, then it implies that e_1 and a_1 must have shared user identifiers prior to the auction. Otherwise, a_1 would have no way of identifying u as the source of the impression during RTB (since the bid request originates from e_1 and not the browser), and would not pay the premium price to win the auction. This is explained in more detail in section § 2.3.

My proposed technique addresses the limitations of prior works as it relies on a methodology that detects information sharing based on causal inferences, rather than relying on HTTP content. Thus, this methodology can defeat obfuscation and can detect server-side information sharing. It also provides strong attribution in information-sharing flows since I record detailed provenance of third-party resource inclusions in web pages using an instrumented version of Chromium [17] (see details in § 4.1.2).

I demonstrate the efficacy of this methodology by conducting extensive experiments on real data. I train 90 *personas* by visiting popular e-commerce sites (§ 4.1), and then crawl major publishers to gather retargeted ads [20, 34]. To record detailed information about the provenance of third-party resource inclusions in web pages (i.e., which resource included which other resources), all crawls were performed using an instrumented version of Chromium [17] that records the *inclusion* *chain* for every resource it encounters. In total, I gather 35,448 chains associated with 5,102 unique retargeted ads (§ 4.1).

Next, I use carefully designed pattern matching rules in § 4.3.1.1 to categorize each of the retargeted ad chains into four different categories, which reveal 1) the pair of A&A companies that shared information in order to serve the retarget, and 2) the mechanism they used to share the data (e.g., cookie matching, server-side matching).

Overall, I found more than 1000 A&A domains in my dataset. These A&A domains consist of trackers, SSPs, exchanges, and various other business models. My methodology also identified 200 cookie matching pairs, out of which 31% were missed by heuristics used by prior works to find cookie matching. Furthermore, I provide empirical evidence that Google shares tracking data across its services by detecting server-side information flows.

Using the methodology described above and the data collected from carefully crafted experiments using retargeted ads, I identify information sharing flows between more than 1,000 A&A companies. However, we are still a key ingredient away from gaining an accurate picture of the privacy landscape under the RTB ecosystem: we need an accurate and comprehensive list of A&A companies which act as ad exchanges during RTB auctions. Ad exchanges play a vital role since within a single ad auction, they can share user tracking data with tens of other A&A companies to solicit bids. Although I can identify ad exchanges from the data we collected, as explained in § 4.3.3, this technique will have both false positives and negatives. Given the "power" ad exchanges possess, we need a more systematic way of identifying ad exchanges.

1.2.2 Transparency & Compliance: An Analysis of the `ads.txt` Standard

Given that we need a list of A&A companies which act as ad exchanges during RTB auctions, I make use of a recently introduced transparency standard called *Authorized Digital Sellers* (`ads.txt`). The `ads.txt` standard was introduced by the Interactive Advertising Bureau (IAB) in 2017 [167]. The motivation behind `ads.txt` is to tackle the issue of *domain spoofing*, which is a form of advertising fraud that has long plagued the RTB ecosystem.

The fundamental issue that enables domain spoofing is the opacity of the RTB ecosystem: DSPs cannot tell which exchanges are *authorized* to sell impression inventory from a given publisher. This lack of transparency gives attackers the ability to spoof inventory from any publisher. `ads.txt` is designed to rectify this transparency problem by allowing publishers to state, in a machine-readable format, which ad exchanges are authorized to sell their impression inventory [83]. To opt-in to the standard, a publisher must place a file named `/ads.txt` at the root of their website; exchanges and advertisers (DSPs) can then download the file and verify the authenticity of bid requests.

`ads.txt` is meant to bring more transparency to the opaque ecosystem of RTB, by making it explicit which third-party domains in a given first-party context are ad exchanges. In aggregate, `ads.txt` data has the potential to reveal, for the first time, the relationships between publishers, ad exchanges, and DSPs. I use this as an opportunity to gather a list of ad exchanges involved in the RTB ecosystem.

To this end, I conduct a 15-month longitudinal, observational study of the `ads.txt` standard on Alexa Top-100K publishers. This data also provides me with a unique opportunity to understand whether ad exchanges and DSPs are complying with the `ads.txt` standard in the effort towards combating domain spoofing, which was the main motivation behind the introduction of the standard. This is an important issue on its own since it is not clear how effective the standard is in the complex RTB ecosystem. Hence, I conduct this study to understand the following two questions:

1. Can the transparency offered by the `ads.txt` standard provide useful data to extract a list of A&A domains that act as ad exchanges?
2. How effective is the `ads.txt` standard at combating domain spoofing? In particular, are ad exchanges and DSPs complying with the standard?

To answer these questions, I crawl `ads.txt` files from Alexa Top-100K websites every month between January 2018 and April 2019. In addition to collecting the `ads.txt` file, I also collect inclusion resources from each website to gather information about the A&A companies that interact with the website. This data allows me to observe whether exchanges and DSPs appear to be in compliance with the rules stipulated in publishers' `ads.txt` files.

With respect to transparency, `ads.txt` files allow us to isolate 1,035 unique domains belonging to ad exchanges from 62% of the Alexa Top-100K publishers that display ads via RTB auctions. That said, I also find that `ads.txt` data has a variety of imperfections, and I develop methods to mitigate these deficiencies. Concerning compliance, I find that the vast majority of RTB ads in our sample were bought from authorized sellers. This suggests that ad exchanges and DSPs are complying with the standard. However, I also see that domain spoofing is still possible because major ad exchanges still accept impression inventory from publishers that have not yet adopted `ads.txt`. Further, I document cases where major ad exchanges purchased impressions from unauthorized sellers, in violation of the standard.

Now that we have a systematic way of identifying ad exchanges, we can use this list, along with the information sharing data gathered from crawled ad inclusion chains to model the privacy digital footprint of web users.

1.2.3 Modeling User's Digital Privacy Footprint

So far I have discussed how the Real Time Bidding ecosystem can affect users' privacy digital footprint. To date, technical limitations and incomplete data have prevented researchers from developing accurate models to demonstrate the privacy implications of RTB as implemented in the modern ad ecosystem. In this dissertation, I have proposed a generic methodology to detect information sharing between arbitrary A&A companies. Additionally, to capture the effect of ad exchanges on privacy leakage during RTB auctions, I use the `ads.txt` standard to systematically identify ad exchanges.

We can use this data to demonstrate the effect of RTB auctions on users' digital footprint. However, due to the enormous complexity of the ad ecosystem and close collaboration among A&A companies, we cannot accurately determine the extent of privacy leakage if we look at RTB auctions in isolation. A natural way to model this complex ecosystem is in the form of a graph. Graph models that accurately capture the relationships between publishers and A&A companies are extremely important for practical applications, such as estimating revenue of A&A companies [74], predicting whether a given domain is a tracker [102], or evaluating the effectiveness of domain-blocking strategies on preserving users' privacy.

To this end, I use the information flows between A&A companies and the list of exchanges to model the advertising ecosystem in the form of a graph called an *Inclusion* graph. By simulating browsing traces for 200 users based on empirical data, I show that the *Inclusion* graph can be used to model the diffusion of user tracking data across the advertising ecosystem.

I demonstrate that due to RTB, the major A&A companies observe the vast majority of users' browsing history. Even under restrictive conditions, where only a small number of well-connected ad exchanges indirectly share impressions during RTB auctions, the top 10% of A&A companies observe more than 91% of impressions and 82% of visited publishers. This is a key result as it highlights that A&A companies observe far greater amounts of user information than what has been demonstrated by prior works [5, 61].

Furthermore, I simulate the effects of ad and tracker blocking on information learned by A&A companies. In particular, I evaluate the following five different blocking strategies:

1. Randomly blocking 30% of the A&A nodes from the *Inclusion* graph.
2. Blocking the top 10% of A&A nodes from the *Inclusion* graph.
3. Blocking all 594 A&A nodes from the Ghostery [73] blacklist.
4. Blocking all 412 A&A nodes from the Disconnect [52] blacklist.
5. Emulating the behavior of AdBlock Plus [7], which is a combination of whitelisting A&A nodes from the Acceptable Ads program [190] and blacklisting A&A nodes from EasyList [54]. After whitelisting, 634 A&A nodes are blocked.

I find that AdBlock Plus (the world's most popular ad-blocking browser extension [122, 159]) is ineffective at protecting users' privacy because major ad exchanges are whitelisted under the Acceptable Ads program [190]. In contrast, Disconnect [52] blocks the most information flows to advertising domains, followed by the removal of top 10% A&A domains. However, the most important observation throughout these experiments is that even with strong blocking methods, major A&A domains still observe 40–70% of user impressions.

1.3 Roadmap

The remainder of this dissertation is organized as follows. In chapter 2, I provide background and introduce key definitions for the online advertising ecosystem. I discuss how online display advertising has moved towards the auction-based RTB ecosystem, and how A&A companies share user identifiers using cookie matching to facilitate RTB. Then, in chapter 3, I provide a detailed overview of related work that has motivated this dissertation.

In chapter 4, I propose a content- and platform-agnostic methodology to detect information sharing between arbitrary A&A companies using retargeted ads. In chapter 5, I provide a longitudinal analysis of the `ads.txt` standard to 1) isolate a list of A&A domains that act as ad exchanges during RTB auctions, and 2) measure adoption and compliance of the standard to combat domain spoofing fraud during RTB auctions. In chapter 6, I use techniques and datasets from my dissertation to build models for determining how user tracking information gets diffused in the advertising ecosystem. I also demonstrate the effectiveness (or lack thereof) of the ad and tracker blocking strategies at preventing leakage of user data to A&A companies.

I conclude by providing a discussion on the findings of this dissertation and future work in chapter 7.

Chapter 2

Background and Definitions

In this chapter, I provide background and definitions about the online advertising ecosystem that will be essential to the rest of this dissertation. I start by describing online display advertising and the different entities involved. Then I discuss how the ecosystem has evolved and moved towards targeted advertising. Finally, I give an overview of Real Time Bidding and describe how cookie matching is done to facilitate displaying of an advertisement through RTB.

2.1 Online Display Advertising

In order to provide free accessibility to their content, *publishers* (e.g., news websites, blogs, etc.) generate revenue by displaying ads on their website from a plethora of advertisers (e.g., *Pepsi*, *Nike* etc.). This symbiotic relationship between the publishers and advertisers is crucial to the sustainability of the modern internet.

Fundamentally, online display advertising is a matching problem. On one side are publishers who produce content, and earn revenue by displaying ads to users. And, on the other side are advertisers who want to display ads to particular users (e.g., based on demographics or market segments). Unfortunately, the online user population is fragmented across hundreds of thousands of publishers, making it difficult for advertisers to reach desired customers. On the flip side, given the vast number of advertisers who want to display ads, it is near impossible for a publisher to maintain business relationships with multiple advertisers to display their ads. This is where *ad networks* step in.

Ad networks bridge this gap by aggregating *inventory* from publishers (i.e., space for displaying ads) and filling it with ads from advertisers. Ad networks make it possible for advertisers to reach

Table 2.1: Key terms used throughout this dissertation.

Term	**Description**
Publisher / Website	Websites /Apps that distribute media to consumers (e.g., `cnn.com`, `weather.com`)
Advertiser	Companies that want to advertise their products to customers (e.g., *Nike*)
Impression	Attention of users (e.g., via page visits)
A&A	Advertising and Analytics related domains
Domain	Effective 2^{nd}-level name (e.g., *doubleclick, openx*)
Supply Side Platform (SSP)	A&A domain that works with publishers to manage their relationships with multiple ad exchanges
Ad Exchange	A&A domain that holds auctions in RTB to solicit bids from DSPs
Demand Side Platform (DSP)	A&A domain that places bids on behalf of advertisers
Privacy Footprint	Browsing history exposed to A&A domains, including domains, URLs, and visit times

a broad swath of users, while also guaranteeing a steady stream of revenue for publishers. While there are several revenue models (e.g., attention reward tokens in Brave's advertising model [178]), inventory is typically sold using a Cost per Mille (CPM) model, where advertisers purchase blocks of 1000 *impressions* (views of ads), or a Cost per Click (CPC) model, where the advertiser pays a small fee each time their ad is clicked by a user.

Over time, the online display ad ecosystem has become more dynamic and has grown tremendously [58]. This has led advertising networks to adapt and specialize in specific roles. For example, while some ad networks work closely with publishers to help maximize their revenue, others collaborate closely with advertisers to help them reach specific audiences [21]. These ad networks participate actively in auctions held by ad exchanges (another specialized role) under the Real Time Bidding model [147]. **I collectively refer to companies engaged in analytics and advertising as *A&A* companies.** (§ 2.3). Mayer et al. presents an accessible introduction to this topic in [123].

Some of the major roles A&A companies have specialized into are:

- **Trackers.** An A&A company that tracks the activity of users across the web by embedding "tracking pixels" or other resources in publishers' web pages.
- **Ad Exchanges.** Implement Real Time Bidding (RTB) auctions to sell impressions to advertisers.
- **Supply Side Platforms (SSPs).** Work closely with publishers to manage their relationships with multiple ad exchanges, to maximize revenue and ensure that all impression inventory is sold.
- **Demand Side Platforms (DSPs).** Work closely with advertisers to assess the value of each impression, optimize bid prices, and implement advertising campaigns.

Note that a single A&A company may play multiple roles in the ecosystem.

Privacy Digital Footprint. While A&A companies specialize in a variety of roles and business models, their overall goal is to serve, or facilitate the serving of, targeted online advertisements. A&A companies achieve this goal by collecting a variety of data about users, including: personally identifiable information, e.g., IP addresses, usernames on websites, email addresses, etc. [60, 110, 165]; hardware and software characteristics from users' devices, potentially to facilitate fingerprinting [61, 168] and reidentification [5, 103]; demographics and preferences from consumer surveys; and behavioral signals gleaned from search keywords, social interactions (e.g., comments and likes), and browsing history (e.g., the links that users click and the URLs they visit).

My work is focused on this last category of data: browsing history. A&A companies have long relied on a variety of tracking techniques to collect the domains and URLs visited by users [5, 61, 109, 110]. Typically, this data is aggregated and used to infer (1) demographic traits and (2) interest profiles about users [20, 24, 86, 113, 114], which in turn are used to target ads. Browsing history is a privacy-sensitive class of data since it may include visits to sensitive destinations, or it may allow third-parties to infer sensitive attributes about a person (e.g., a health condition, sexual and political orientation, a desire to quit a job, etc.).

In my work, I refer to the browsing history learned by A&A companies as the *privacy digital footprint* of the user. In particular, this consists of the browsing activity on the web observed by the A&A companies, which includes dates and times of domains and URLs visited by a person. As I note above, there are other key aspects of user privacy that are out of the scope of this dissertation. For example, understanding the type of information leaked (e.g., gender, sexual orientation, phone numbers, *etc.*) to A&A companies is important to understand, but requires deeper inspection of the traffic flows under controlled experiments than I have performed. Table 2.1 contains some of the key terms I use throughout this dissertation.

2.2 Targeted Advertising

Initially, the online display ad industry focused on generic brand ads (e.g., "Enjoy Coca-Cola!") or *contextual ads* (e.g., an ad for Microsoft on StackOverflow). However, the industry quickly evolved towards *behavioral targeted ads* that are served to specific users based on their browsing history, interests, and demographics.

2.2.1 Online Tracking

With the web becoming more personalized, ad networks have adapted over time to show relevant and more personalized content to end-users. However, in order to provide personalized content, ad networks must collect information about users (to infer their interests). They do so by tracking users across the web and observing their browsing history using third-party cookies [33, 107, 109, 168] and fingerprinting techniques [6,56,61,103,104,133,138,141,182]. Fo example, publishers embed JavaScript or invisible "tracking pixels" that are hosted by tracking companies into their web pages, thus any user who visits the publisher also receives third-party cookies from the tracker (I discuss other tracking mechanisms in greater depth in § 3.2).

Numerous studies have shown that trackers are pervasive across the web [33, 107, 109, 168], which allows A&A companies to collect users' browsing history. All major ad exchanges, like DoubleClick and Rubicon, perform user tracking, but there are also entities like Oracle BlueKai that just specialize in tracking.

The amount of collected information enables A&A companies to show targeted ads to users. Depending on the amount of information, the targeting can become very specific. The tracking information also helps DSPs make *bidding* decision during RTB, where they bid high or low for user impressions, depending on the amount of information they have about the user [147].

2.2.2 Retargeted Ads

The methodology proposed in this dissertation uses *retargeted ads* as a tool to detect information sharing between A&A companies. Retargeted ads are the most specific form of targeted display ads. Two conditions must be met for a DSP to serve a retargeted ad to a user u: 1) the DSP must know that u browsed a specific product on a specific e-commerce site, and 2) the DSP must be able to uniquely identify u during an RTB auction. If these conditions are met, the DSP can serve u a highly personalized ad, reminding them to purchase the product from the retailer. Cookie matching (explained in the next section) is crucial for ad retargeting since it enables DSPs to meet requirement (2).

2.3 Real Time Bidding

Over time, the mechanisms for selling and buying *impressions* have become *programmatic* via *Real Time Bidding (RTB) auctions*. In industry parlance, *publishers* aim to monetize their *impres-*

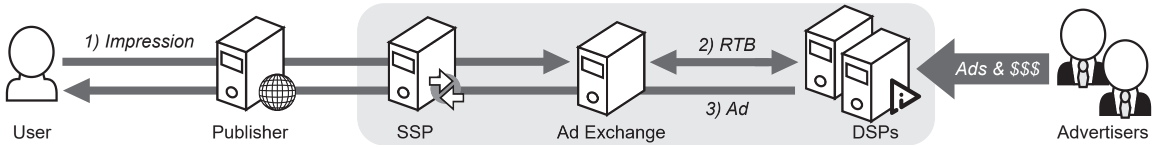


Figure 2.1: The display advertising ecosystem. Impressions and tracking data flow left-to-right, while revenue and ads flow right-to-left.

sion inventory (i.e., the attention of people visiting their service) by selling it to advertisers.

2.3.1 Overview

At a high-level, whenever a person visits a publisher, their browser will contact an *ad exchange* that serves as the auctioneer. The ad exchange solicits bids for the impression from DSPs, who have just milliseconds to submit bids on behalf of advertisers. The ad exchange then redirects the user's browser to the winning DSP, so they may serve an ad. It is estimated that programmatic advertising will account for 83% of all US digital display advertising by 2020 [58]. RTB is popular because it increases fluidity in the advertising market, as well as allowing publishers to increase their revenue (in theory) by selling their inventory to the highest bidders on-demand. Figure 2.1 shows how impressions, user tracking data, and revenue flow across various entities involved in RTB auctions.

Although RTB auctions are conceptually simple, they are complex in practice. With respect to the *sell-side*, publishers form business relationships with ad exchanges and other *Supply-Side Platforms (SSPs)* that facilitate the selling of impressions. Examples of ad exchanges include the Google Marketing Platform (formerly Doubleclick), Rubicon, and OpenX. With respect to the *buy-side*, *Demand-Side Platforms (DSPs)* represent advertisers by purchasing impressions to implement their campaigns. Examples of DSPs include Criteo, Quantcast, and MediaMath. Note that many companies offer seller- and buyer-side products (e.g., Google and Rubicon), complicating their role in the ecosystem. Furthermore, impressions can be resold after they are won, i.e., the winner of an RTB auction may be another ad exchange, which will then hold another auction, etc. This can lead to long *chains* of transactions that separate the true source of an impression from the DSP that eventually serves an ad. This complexity enables various forms of advertising fraud, such as domain spoofing, which is a topic I will return to in chapter 5.

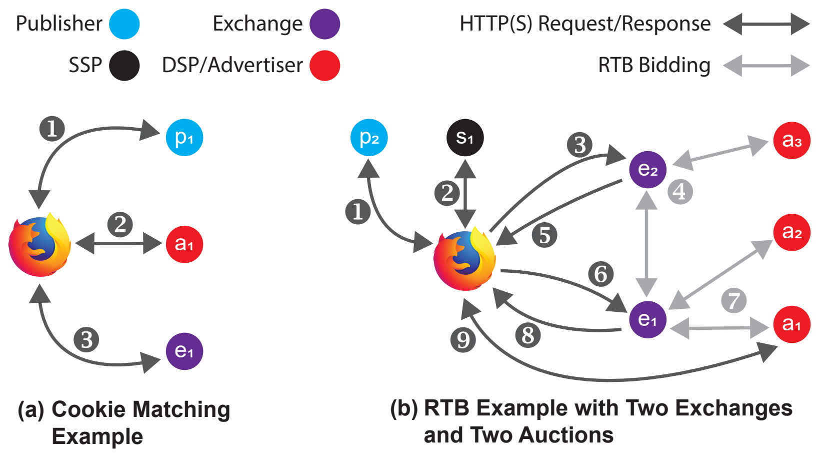


Figure 2.2: Examples of **(a)** cookie matching and **(b)** showing an ad to a user via RTB auctions. **(a)** The user visits publisher p_1 ❶ which includes JavaScript from advertiser a_1 ❷. a_1's JavaScript then cookie matches with exchange e_1 by programmatically generating a request that contains both of their cookies ❸. **(b)** The user visits publisher p_2, which then includes resources from SSP s_1 and exchange e_2 ❶–❸. e_2 solicits bids ❹ and sells the impression to e_1 ❺ ❻, which then holds another auction ❼, ultimately selling the impression to a_1 ❽ ❾.

2.3.2 Cookie Matching

During RTB, an ad exchange holds an auction and DSPs submit bids for user impressions. The amount of money that a DSP bids on a given impression is intrinsically linked to the amount of information they have about that user. For example, a DSP is unlikely to bid highly for user u whom they have never observed before, whereas a DSP may bid heavily for user v whom they have recently observed browsing high-value websites (e.g., the baby site TheBump.com).

However, the Same Origin Policy (SOP) hinders the ability of DSPs to identify users in ad auctions. As shown in Figure 2.1, requests are first sent to an SSP which forwards the impression to an exchange. At this point, the SSP's and exchange's cookies are known, but not the DSPs' cookies. This leads to a catch-22 situation: a DSP cannot read its cookies until it contacts the user, but it cannot contact the user without first bidding and winning the auction.

To circumvent SOP restrictions, ad exchanges and DSPs engage in *cookie matching* (sometimes called *cookie syncing*). Figure 2.2(a) illustrates the typical process used by A&A companies to match cookies. When a user visits a website ❶, JavaScript code from a third-party ad network a_1 is automatically downloaded and executed in the user's browser ❷. This code may set a cookie in the user's browser, but this cookie will be unique to a_1, i.e., it will not contain the same unique identifiers as the cookies set by any other A&A companies. Furthermore, as mentioned above, SOP restrictions prevent a_1's code from reading the cookies set by any other domain. To facilitate bidding in future RTB auctions, a_1 syncs their identifiers with those set by an ad exchange like e_1. As shown in the figure, a_1's JavaScript accomplishes this by programmatically causing the browser to send a request (via HTTP redirect) to e_1 ❸. The JavaScript includes a_1's cookie in the request, and the browser automatically adds a copy of e_1's cookie, thus allowing e_1 to create a match between its cookie and a_1's[1]. In the future, if a_1 participates in an auction held by e_1, it will be able to identify the user using a previously matched cookie. Note that some ad exchanges (including DoubleClick) send cryptographically hashed cookies to their partners, which prevents the ad network's true cookies from leaking to third-parties.

2.3.3 Advertisement Served via RTB

Figure 2.2(b) shows an example of how an ad may be shown on publisher p_2 using RTB auctions. When a user visits p_2 ❶, JavaScript code is automatically downloaded and executed either from a *Supply Side Platform (SSP)* ❷ or an ad exchange. Eventually the impression arrives at the auction held by ad exchange e_2 ❸, and e_2 solicits bids from DSPs ❹. Note that **all participants in the auction observe the impression**; however, because only e_2's cookie is available at this point, auction participants that have not matched cookies with e_2 will not be able to identify the user.

The process of filling an impression may continue even after an RTB auction is won, because the winner may be yet another ad exchange or ad network. As shown in Figure 2.2(b), the impression is purchased from e_2 by e_1 ❺ ❻, who then holds another auction ❼ and ultimately sells to a_1 (the advertiser from the cookie matching example) ❽ ❾. Ad exchanges and DSPs routinely match cookies with each other to facilitate the flow of impression inventory between markets.

[1] Cookie matching can happen in both directions, i.e., from a_1 to e_1 and from e_1 to a_1.

Chapter 3

Related Work

In this chapter, I present the related work that has motivated and informed this dissertation. I begin by providing an overview of general studies on the display advertising ecosystem. Then, I survey the related work documenting the pervasiveness of online tracking, tracking mechanisms used by A&A companies, user perceptions regarding online tracking, and ongoing efforts to make the ecosystem more transparent. I conclude this chapter by discussing related work that has specifically examined RTB and cookie matching, and motivate the need for this dissertation by highlighting their limitations.

3.1 The Online Advertising Ecosystem

Numerous studies have chronicled the online advertising ecosystem, which is composed of companies that track users, serve ads, act as platforms between *publishers* and advertisers, or all of the above. Mayer et al. presents an accessible introduction to this topic in [123].

Guha et al. [86] were the first to develop a controlled and systematic methodology based on trained *personas* to measure online ads on the web. Their work has been very influential in subsequent studies, including this dissertation. Barford et al. [20] take a much broader look at the *adscape* to determine who the major ad networks are, what fraction of ads are targeted, and what user characteristics drive targeting. Carrascosa et al. [34] take an even finer-grained look at targeted ads by training *personas* that embody specific interest profiles (e.g., cooking, sports), and find that advertisers routinely target users based on sensitive attributes (e.g., religion).

In chapter 4, I make use of *personas* along with retargeted ads to detect information sharing between A&A companies. **None** of these studies mentioned above examine retargeted ads; Carrascosa et al. specifically excluded retargets from their analysis [34].

There has been work in the space of the mobile ad ecosystem as well. Rodriguez et al. [189] were one of the first to measure the ad ecosystem on mobile devices. More recently, Razaghpanah et al. [162] presented insights into the mobile advertising and tracking ecosystem. Using real-world mobile traffic data, they discovered 2,121 A&A services and analyze their business relationships with one another. Although my dissertation does not focus on the mobile ad space, the proposed methodologies can be extended to other platforms (see chapter 7).

Researchers have found that the information and revenue in the ad ecosystem are skewed towards top players. Gill et al. [74] used browsing traces to study the economy of online advertising and discovered that the revenues are skewed towards the largest trackers (primarily Google). More recently, Cahn et al. [33] performed a broad survey of cookie characteristics across the web and found that $<1\%$ of trackers can aggregate information across 75% of websites in the Alexa Top-10K. Englehardt et al. [61] discovered similar results in their analysis of Alexa Top-1M websites. In particular, they found that 12 out of the top 20 third-party domains belong to Google. Falahrastegar et al. [64] looked at third-party prevalence across geographic regions.

Researchers have also studied malicious and bad practices in the advertising ecosystem. Zarras et al. [198] studied malicious ad campaigns and the ad networks associated with them, whereas in my prior work I found that some advertisers were not following industry guidelines and were serving poor quality ads [23].

3.2 Online Tracking

To facilitate ad targeting, participants in the ad ecosystem must extensively track users. In this section, I survey the related work that identifies tracking mechanisms employed by A&A domains and proposes solutions to combat online tracking.

3.2.1 Tracking Mechanisms

Krishnamurthy et al. were one of the first to bring attention to the pervasiveness of trackers and their privacy implications for users [109], and since then they have been cataloging the spread of trackers and assessing the ensuing privacy implications [106–108]. Recently, Lerner et al. [117] examined the evolution of third-party trackers from 1996-2016.

Ad networks have evolved their tracking techniques over time, sometimes going to extraordinary lengths to collect and retain user information. Roesner et al. [168] developed a comprehensive taxonomy of different tracking mechanisms that store state in users' browsers (e.g., cookies, HTML5 LocalStorage, and Flash LSOs), as well as strategies to block them. Li et al. [118] show that most tracking cookies can be automatically detected using simple machine learning methods.

Although users can try to evade trackers by clearing their cookies or using private/incognito browsing modes, companies have fought back using techniques like *Evercookies* and *fingerprinting*. Evercookies store the tracker's state in many places within the browser (e.g., FlashLSOs, Etags, *etc.*), thus facilitating the regeneration of tracking identifiers even if users delete their cookies [18, 103, 125, 179].

Fingerprinting involves generating a unique ID for a user based on the characteristics of their browser [56, 133, 138], browsing history [146], browser extensions [182] and computer (e.g., the HTML5 canvas [134]). Recently, Englehardt et al. [61] found trackers fingerprinting users via the JavaScript `Audio` and `Battery Status` APIs. Several studies have found trackers in-the-wild that use fingerprinting techniques [6, 61, 104, 141]; Nikiforakis et al. [140] proposed techniques to mitigate fingerprinting by carefully and intentionally adding more entropy to users' browsers.

Researchers have also studied the state of tracking and its privacy implications on mobile devices [27, 57, 82, 162, 165, 189]. They have noticed that tracking is ubiquitous on mobile devices and that apps use embedded sensors (e.g., camera, microphone, GPS) to extensively track users. Additionally, there have been two prominent studies on cross-device tracking. Brookman et al. [30] from the Federal Trade Commission (FTC) surveyed 100 popular websites to study the potential for cross-device tracking, although they did not measure the actual prevalence of cross-device tracking. In contrast, Zimmeck et al. [201] found empirical evidence of cross-device tracking in their survey of 126 internet users.

3.2.2 Users' Perceptions of Tracking

Various surveys have found that people have concerns about the amount and type of information collected about them. McDonald et al. reported that 64% of the participants they surveyed found targeted advertising to be invasive [124]. Similarly, Turow et al. found that the majority of Americans feel that they do not have a meaningful choice with respect to the collection and use of their data by third-parties; thus, respondents were resigned to giving up their data [187]. Peoples' feelings about lack of agency may be rooted, in part, by widespread misconceptions about how targeted advertising systems are implemented [11, 121]. Balebako et al. discussed user concerns regarding behavioral advertising and evaluated the effectiveness of privacy tools as counter mechanism [19].

Studies have found that a variety of factors influence people's perceptions of tracking and online advertising. Ur et al. reported that people found targeted advertising to be both useful and privacy-invasive depending on how much they trusted the advertising company [188]. Similarly, Leon et al. surveyed 2,912 participants and found that they were willing to share information with advertisers if they were given more control over what was shared and with whom [116]. Like Ur et al., O'Donnell et al. surveyed 256 participants and found targeted advertising to be useful under a variety of circumstances (e.g., ads around major life events) [143]. However, Plane et al. found that people were very concerned when ad targeting resulted in discrimination (e.g., by targeting racial attributes) [156]. These findings highlight the complexity of peoples' relationship with target advertising, i.e., how trust, context, content, control, and effect commingle to shape perceptions of individual advertisements and the industry as a whole.

Dolin et al. surveyed people to understand how ad *explanations* (small disclosures near advertisements that provide insight into how the ad was targeted) impact peoples' opinions of targeted advertising. They found that peoples' comfort level varied based on the explanation they were given for how the targeted interest was inferred [53]. They also report that the accuracy of the inferred interests was strongly, positively correlated with user comfort, regardless of the sensitivity of the interest. In my prior work, I shed light on the accuracy of inferred interests [24]. I find that user interest profiles contain noisy data and low-relevance interests. In particular, the 220 users I surveyed found the majority of their interests as not relevant (only 27% strongly relevant), and don't consider ads targeted to low-relevance interests to be useful.

3.2.3 Blocking & Anti-Blocking

On one hand, tracking has enabled advertisers to show relevant ads to users, while on the other, it has raised concerns among users about the amounts and types of information being collected about them [124, 187]. To avoid pervasive tracking, users are increasingly adopting tools that block trackers and ads [122, 159]. There has also been a development towards whitelisting "acceptable" ads [190]. Merzdovnik et al. [127] and Iqbal et al. [99] performed large scale measurements of blocking extensions and techniques to determine which are most effective.

Concerned with the increased adoption of ad and tracker blocking tools, advertisers have started developing techniques to counter them. Merzdovnik et al. [127] critically examined the effectiveness of tracker blocking tools; in contrast, Nithyanand et al. [142] studied advertisers' efforts to counter ad blockers. Mughees et al. [137] examined the prevalence of anti-ad blockers in the wild.

Recently, advertisers were reported by user communities for displaying ads (even with ad blockers installed) through WebSockets and WebRTC [91, 166]. Similarly, WebRTC has also been known to reveal user IP addresses [61, 164]. Snyder et al. [176] performed a browser feature usage survey and showed that ad and tracking blocking extensions do not block all standards equally, with WebSockets being blocked 65% of the times. Franken et al. [71] reported that blocking extensions could sometimes be bypassed using WebSockets. They found that the extention developers made the mistake of using "`http://*, https://*`" filters instead of "`ws://*, wss://*`" for the `onBeforeRequest` event, which prevents the interception of WebSocket connections. In my prior work [22], I shed light on A&A companies that were circumventing ad blockers through WebSockets to track users and display ads.

The research community has proposed a variety of mechanisms to stop online tracking that goes beyond blacklists of domains and URLs. Li et al. [118] and Ikram et al. [96] used machine learning to identify trackers; Papaodyssefs et al. [150] proposed the use of private cookies to mitigate tracking; Nikiforakis et al. [140] added entropy to the browser to combat fingerprinting. More recently, Zhu et al. [200] and Iqbal et al. [100] proposed machine learning based approaches to automatically block trackers and advertisements. However, despite these efforts, third-party trackers are still pervasive and pose real privacy issues to users [127].

3.2.4 Transparency

In an effort to make the advertising ecosystem more transparent, some advertising companies (e.g., Google, Facebook) have built transparency tools called Ad Preference Managers (APMs) to enable users to see, and in some cases modify, what information has been inferred about them. However, studies have highlighted certain issues with these tools: they lack coverage [14, 194], exclude sensitive user attributes [48], and infer noisy and irrelevant interests [24, 50, 186].

Several studies specifically focus on tracking data collected by Google, since their trackers are more pervasive than any others on the web [33, 74]. Alarmingly, two studies have found that Google's Ad Preferences Manager, which is supposed to allow users to see and adjust how they are being targeted for ads, actually hides sensitive information from users [48, 194]. This finding is troubling given that several studies rely on data from the Ad Preferences Manager as their source of ground-truth [20, 35, 86]. To combat this lack of transparency, Lecuyer et al. [113, 114] have built systems that rely on controlled experiments and statistical analysis to infer the profiles that Google constructs about users. Castelluccia et al. [35] go further by showing that adversaries can infer users' profiles by passively observing the targeted ads they are shown by Google.

3.3 Real Time Bidding and Cookie Matching

As I note in chapter 2, A&A companies have to perform *cookie matching* to be able to participate in RTB auctions. Although ad networks have been transitioning to RTB auctions since the mid-2000s, there have been only a handful of empirical studies that have examined cookie matching.

Acar et al. [5] found that hundreds of domains passed unique identifiers to each other while crawling websites in the Alexa Top-3K. Falahrastegar et al. [65] examine the clusters of domains that all share unique, matched cookies using crowdsourced browsing data. Additionally, Ghosh et al. use game theory to model the incentives for ad exchanges to match cookies with their competitors, but they provide no empirical measurements of cookie matching [72].

Olejnik et al. [147] noticed that ad auctions were leaking the winning bid prices for impressions, thus enabling a fascinating behind-the-scenes look at RTB auctions. In addition to examining the monetary aspects of auctions, Olejnik et al. found 125 ad exchanges using cookie matching. Papadopoulos et al. [149] ran their own ad campaigns to develop a model that can collect bid prices even when they are encrypted.

Furthermore, some studies have also examined retargeted ads, which are directly facilitated by cookie matching and RTB. Liu et al. [119] identified and measured retargeted ads served by DoubleClick by relying on unique AdSense tags that were embedded in ad URLs. Olejnik et al. [147] crawled specific e-commerce sites to elicit retargeted ads from those retailers and observed that retargeted ads could cost advertisers over $1 per impression (an enormous sum, considering contextual ads sell for <$0.01).

Limitations. Although prior studies provide insights into the widespread practice of *cookie matching*, they have significant methodological limitations, which prevent them from observing *all* forms of information sharing between A&A companies. Specifically:

1. **Resource Attribution:** These studies cannot determine the precise information flows between ad exchanges, i.e., which parties are sending or receiving information [5]. The fundamental problem is that HTTP requests, and even the DOM tree itself, do not reveal the true sources of resource inclusions in the presence of dynamic code (JavaScript, Flash, *etc.*) from

third-parties. For example, a script from t1.com embedded in pub.com may share identifiers with t2.com using dynamic AJAX, but the Referer appears to be pub.com, thus potentially hiding t1's role as the source of the flow (see § 4.1.2).

2. **Obfuscation:** These studies rely on locating unique user IDs that are transmitted to multiple third-party domains [5, 65, 147]. Unfortunately, this will miss cases where exchanges send permuted or obfuscated IDs to their partners. Indeed, DoubleClick is known to do this [2]. The two studies that have examined the behavior of DoubleClick have done so by relying on specific cookie keys and URL parameters [119, 147]. This is not a robust way of performing detection since in the future DoubleClick can change the parameter names.
3. **Server-Side Matching:** Since these methods rely on analyzing HTTP content, they will miss information sharing that happens on the server-side (without ever going through the user's browser). For example, two ad networks that belong to the same parent company can share user tracking data with each other without cookie matching. In § 4.3.1.2, I highlight that Google services share identifiers on the server-side.

In general, these limitations stem from a reliance on analyzing specific *mechanisms* for cookie matching. In this dissertation, one of my primary goals is to develop a methodology for detecting cookie matching (and thus, information sharing) that is agnostic to the underlying matching mechanism and instead relies on the fundamental *semantics* of how ad exchanges work under RTB.

Chapter 4

Tracing Information Flows Between A&A companies Using Retargeted Ads

Real Time Bidding (RTB) is quickly becoming the dominant mechanism for buying and selling advertising inventory from publishers [1, 4]. The rise of RTB has forced advertising companies to collaborate more closely with one another. To be able to participate in RTB auctions, A&A domains routinely share user identifiers with each via *cookie matching*, which is a pre-requisite for RTB participation. Despite user concerns about their digital footprint, we currently lack the tools to fully understand how much and how often information is being shared between A&A domains.

Although prior empirical works have relied on heuristics that look for specific strings in HTTP messages to identify flows between ad networks [5,65,147], these heuristics are brittle in the face of obfuscation: for example, DoubleClick cryptographically hashes their cookies before sending them to other ad networks [2]. More fundamentally, analysis of *client-side* HTTP messages is insufficient to detect *server-side* information flows between A&A domains.

In this chapter, I develop a methodology that can detect client- and server-side flows of information between arbitrary A&A domains using retargeted ads. Retargeted ads are the most specific form of behavioral ads, where a user is targeted with ads related to the exact products she has previously browsed (see § 2.3). For example, Bob visits `nike.com` and browses for running shoes but decides not to purchase them. Bob later visits `cnn.com` and sees an ad for the exact same running shoes from Nike.

The key insight is to leverage retargeted ads as a mechanism for identifying information flows. This is possible because the strict conditions that must be met for a retarget to be served allow us to infer the precise flow of tracking information that facilitated the serving of the ad. Intuitively, this methodology works because it relies on the *semantics* of how exchanges serve ads, rather than focusing on specific cookie matching *mechanisms*.

To demonstrate the efficacy of this methodology, I conduct extensive experiments on real data. I train 90 *personas* by visiting popular e-commerce sites, and then crawl major publishers to gather retargeted ads [20, 34]. My crawler is an instrumented version of Chromium that records the *inclusion chain* for every resource it encounters [17], including 35,448 chains associated with 5,102 unique retargeted ads. I use carefully designed pattern matching rules to categorize each of these chains, which reveal 1) the pair of A&A domains that shared information in order to serve the retarget, and 2) the mechanism used to share the data (e.g., cookie matching, server-side matching, *etc.*).

In summary, in this chapter I make the following contributions:

- I present a novel methodology for identifying information flows between A&A domains that is content- and platform-agnostic. This methodology allows us to identify four different categories of information sharing between A&A domains, of which cookie matching is one.
- Using crawled data, I show that the heuristics used by prior works to analyze cookie matching are unable to identify 31% of A&A domain pairs that share data.
- Although it is known that Google's privacy policy allows it to share data between its services [79], I provide the first empirical evidence that Google uses this capability to serve retargeted ads.
- Using graph analysis, I show how the data collected in this study can be used to automatically infer the roles played by different A&A companies (e.g., Supply-Side and Demand-Side Platforms). These results expand upon prior work [77] and facilitate a more nuanced understanding of the online ad ecosystem.

4.1 Methodology

In this chapter, my primary goal is to develop a methodology for detecting flows of user data between arbitrary A&A domains. This includes client-side flows (i.e., cookie matching), as well as server-side flows. In this section, I discuss the methods and data I use to meet this goal. First, I briefly sketch my high-level approach and discuss key enabling insights. Second, I introduce the instrumented version of Chromium that I use during my crawls. Third, I explain how I designed and trained shopper *personas* that view products on the web, and finally I detail how I collected ads using these trained personas.

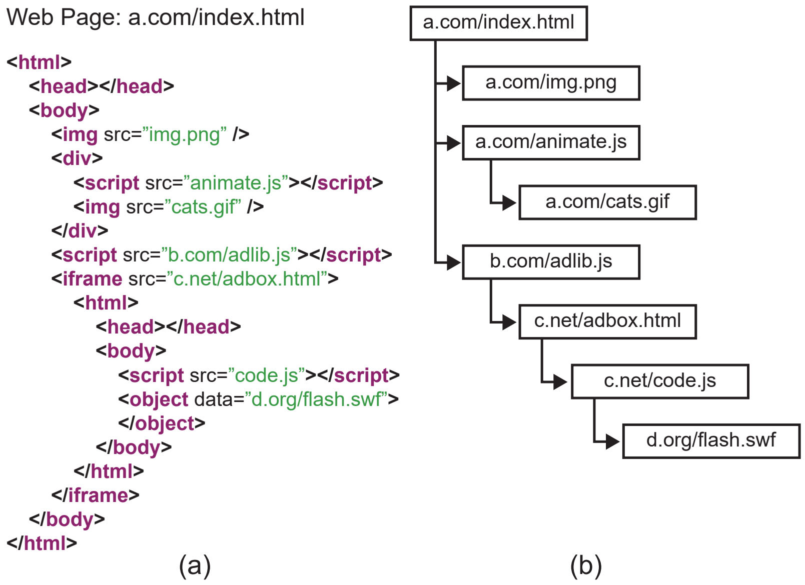
4.1.1 Insights and Approach

Although prior work has examined information flow between A&A companies, these studies are limited to specific types of cookie matching that follow well-defined patterns (see § 3.3). To study arbitrary information flows in a mechanism-agnostic way, I need a fundamentally different methodology.

I solve this problem by relying on a key insight: in most cases, if a user is served a retargeted ad, this proves that ad exchanges shared information about the user (see § 4.3.1.1). To understand this insight, consider the two pre-conditions that must be met for user u to be served a retarget ad for $shop$ by DSP d. *First*, either d directly observed u visiting $shop$, or d must be told this information by SSP s. If this condition is not met, then d would not pay the premium price necessary to serve u a retarget. *Second*, if the retarget was served from an ad auction, SSP s and d must be sharing information about u. If this condition is not met, then d would have no way of identifying u as the source of the impression (see § 2.3).

In this study, I leverage this observation to reliably infer information flows between SSPs / exchanges and DSPs, regardless of whether the flow occurs client- or server-side. The high-level methodology is quite intuitive: have a clean browser visit specific e-commerce sites, then crawl publishers and gather ads. If I observe retargeted ads, I know that ad exchanges tracking the user on the *shopper-side* are sharing information with exchanges serving ads on the *publisher-side*. Specifically, this methodology uses the following steps:

- § 4.1.2: I use an instrumented version of Chromium to record *inclusion chains* for all resources encountered during my crawls [17]. These chains record the precise origins of all resource requests, even when the requests are generated dynamically by JavaScript or Flash. I use these chains in § 4.3 to categorize information flows between ad exchanges.
- § 4.1.3: To elicit retargeted ads from ad exchanges, I design *personas* (to borrow terminology from [20] and [34]) that visit specific e-commerce sites. These sites are carefully chosen to cover different types of products and include a wide variety of common trackers.
- § 4.1.4: To collect ads, each created persona crawl 150 publishers from the Alexa Top-1K list.



Web Page: a.com/index.html

```
<html>
  <head></head>
  <body>
    <img src="img.png" />
    <div>
      <script src="animate.js"></script>
      <img src="cats.gif" />
    </div>
    <script src="b.com/adlib.js"></script>
    <iframe src="c.net/adbox.html">
      <html>
        <head></head>
        <body>
          <script src="code.js"></script>
          <object data="d.org/flash.swf">
          </object>
        </body>
      </html>
    </iframe>
  </body>
</html>
```

Figure 4.1: (a) DOM Tree, and (b) Inclusion Tree.

- § 4.2: I leverage well-known filtering techniques and crowdsourcing to identify retargeted ads from the corpus of 571,636 unique crawled images.

4.1.2 Instrumenting Chromium

Before I can begin crawling, I first need a browser that is capable of recording detailed information about the provenance of third-party resource inclusions in web pages. Recall that prior work on cookie matching was unable to determine which ad exchanges were syncing cookies in many cases because the analysis relied solely on the contents of HTTP requests [5, 65] (see § 3.3). The fundamental problem is that HTTP requests, and even the DOM tree itself, do not reveal the true sources of resource inclusions in the presence of dynamic code (JavaScript, Flash, *etc.*) from third-parties.

To understand this problem, consider the example DOM tree for `a.com/index.html` in Figure 4.1(a). Based on the DOM, we might conclude that the chain $a \rightarrow c \rightarrow d$ captures the sequence of inclusions leading from the root of the page to the Flash object from `d.org`.

However, the direct use of a web page's DOM is misleading because the DOM does not reliably record the inclusion relationships between resources in a page. This is due to the ability of JavaScript to manipulate the DOM at run-time, i.e., by adding new inclusions dynamically. As such, while the DOM is a faithful syntactic description of a webpage *at a given point in time*, it cannot be relied upon to extract relationships between included resources. Furthermore, analysis of HTTP request

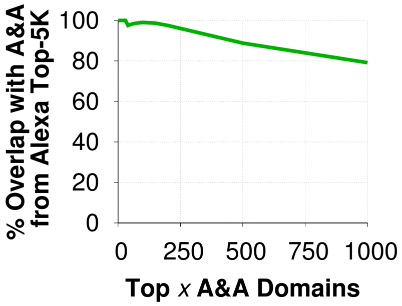


Figure 4.2: Overlap between frequent A&A domains and A&A domains from Alexa Top-5K.

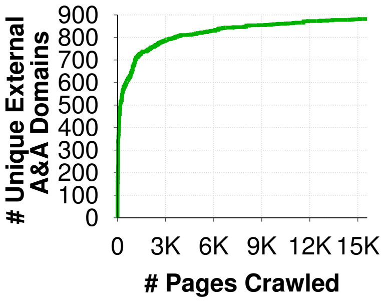


Figure 4.3: Unique A&A domains contacted by each A&A domain as we crawl more pages.

headers does not solve this problem, since the `Referer` is set to the first-party domain even when inclusions are dynamically added by third-party scripts.

To solve this issue, I make use of a heavily instrumented version of Chromium that produces *inclusion trees* directly from Chromium's resource loading code [17]. Inclusion trees capture the semantic inclusion structure of resources in a web page (i.e., which objects cause other objects to be loaded), unlike DOM trees which only capture syntactic structures. The instrumented Chromium accurately captures relationships between elements, regardless of where they are located (e.g., within a single page or across frames) or how the relevant code executes (e.g., via an inline `<script>`, `eval()`, or an event handler). More details about the inclusion trees and how the Chromium binary is instrumented can be found in [17].

Figure 4.1(b) shows the inclusion tree corresponding to the DOM tree in Figure 4.1(a). From the inclusion tree, we can see that the true *inclusion chain* leading to the Flash object is $a \rightarrow b \rightarrow c \rightarrow c \rightarrow d$, since the `iframe` and the Flash are dynamically included by JavaScript from `b.com` and `c.net`, respectively.

Using inclusion chains, I can precisely analyze the provenance of third-party resources included in web pages. In § 4.3, I use this capability to distinguish client-side flows of information between A&A domains (i.e., cookie matching) from server-side flows.

4.1.3 Creating Shopper Personas

Now that I have a robust crawling tool, the next step in the methodology is designing shopper personas. Each persona visits products on specific e-commerce sites, in the hope of seeing retargeted ads when I crawl publishers.

Since we do not know a priori which e-commerce sites are conducting retargeted ad campaigns, these personas must cover a wide variety of sites. To facilitate this, I leverage the hierarchical categorization of e-commerce sites maintained by Alexa[1]. Although Alexa's hierarchy has 847 total categories, there is a significant overlap between categories. I manually selected 90 categories to use for these personas so that they have minimal overlap, as well as cover major e-commerce sites (e.g., Amazon and Walmart) and shopping categories (e.g., sports, jewelry, and baby products).

For each persona, I included the top 10 e-commerce sites in the corresponding Alexa category. In total, the personas cover 738 unique websites. Furthermore, I manually selected 10 product URLs on each of these websites. Thus, each persona visits 100 product URLs across 10 e-commerce sites.

Sanity Checking. The final step in designing these personas is ensuring that the e-commerce sites are embedded with a representative set of trackers. If they are not, then we will not be able to collect targeted ads when we crawl publishers.

Figure 4.2 plots the overlap between the trackers we observe on the Alexa Top-5K websites, compared to the top x trackers (i.e., most frequent) we observe on the e-commerce sites. We see that 84% of the top 100 e-commerce trackers are also present in the trackers on Alexa Top-5K sites[2]. These results demonstrate that our shopping personas will be seen by the vast majority of major trackers when they visit our 738 e-commerce sites.

4.1.4 Collecting Ads

In addition to selecting e-commerce sites for our personas, I must also select publishers to crawl for ads. To this end, I manually select 150 publishers by examining the Alexa Top-1K websites and filtering out those which do not display ads, are non-English, are pornographic, or require logging-in to view the content (e.g., Facebook). I randomly selected 15 URLs on each publisher to crawl (including the homepage).

At this point, I am ready to crawl ads. I initialized 91 copies of our instrumented Chromium binary: 90 corresponding to our shopper personas, and one which serves as a control. During each *round* of crawling, the personas visit their associated e-commerce sites, then visit the 2,250 publisher URLs (150 publishers $*$ 15 pages per publisher). The control *only* visits the publisher URLs, i.e., it does not browse e-commerce sites, and therefore should never be served retargeted ads. The crawlers are executed in tandem, so they visit the publishers URLs in the same order at the

[1] http://www.alexa.com/topsites/category/Top/Shopping

[2] I separately crawled the resources included by the Alexa Top-5K websites in January 2015. For each website, I visited 6 pages and recorded all the requested resources.

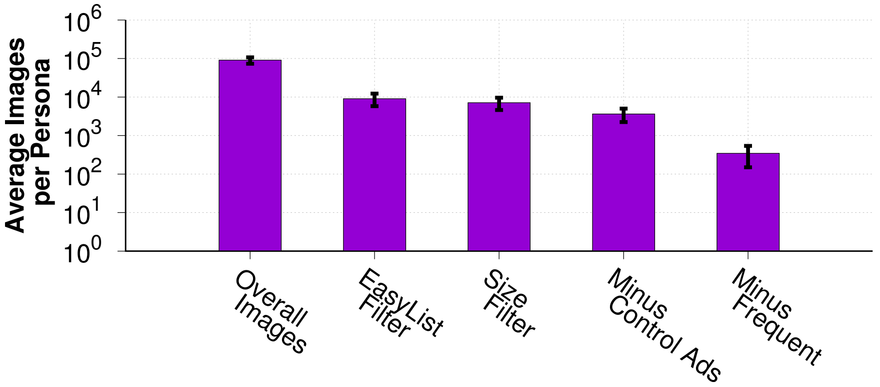


Figure 4.4: Average number of images per persona, with standard deviation error bars.

same time. I hard-coded a 1-minute delay between subsequent page loads to avoid overloading any servers, and to allow time for the crawler to automatically scroll to the bottom of each page. Each round takes 40 hours to complete.

I conducted nine rounds of crawling between December 4 to 19, 2015. I stopped after 9 rounds because I observed that I only gathered 4% new images during the ninth round. The crawlers recorded inclusion trees, HTTP request and response headers, cookies, and images from all pages. At no point did the crawlers click on ads, since this can be construed as click-fraud (i.e., advertisers often have to pay each time their ads are clicked, and thus automated clicks drain their advertising budget). All crawls were done from *Northeastern University's* IP addresses in Boston.

4.2 Image Labeling

Using the methodology in § 4.1.4, I collected 571,636 unique images in total. However, only a small subset are retargeted ads, which are of interest. In this section, I discuss the steps I used to filter down our image set and isolate retargeted ads, beginning with standard filters used by prior work [20, 118], and ending with crowdsourced image labeling.

4.2.1 Basic Filtering

Prior work has used a number of techniques to identify ad images from crawled data. First, I leverage the *EasyList* filter [54][3] provided by *AdBlock Plus* [7] to detect images that are likely to be ads [20, 118]. In our case, I look at the inclusion chain for each image and filter out those in which

[3] https://easylist-downloads.adblockplus.org/easylist.txt

none of the URLs in the chain are a hit against EasyList. This reduces the set to 93,726 unique images.

Next, I filter out all images with dimensions $< 50 \times 50$ pixels. These images are too small to be ads; most are 1×1 tracking pixels.

The final filter relies on a unique property of retargeted ads: they should only appear to personas that visit a specific e-commerce site. In other words, an ad that was shown to our control account (which visits no e-commerce sites) is either untargeted or contextually targeted and can be discarded. Furthermore, any ad shown to >1 persona may be behaviorally targeted, but it cannot be a retarget, and is therefore filtered out[4].

Figure 4.4 shows the average number of images remaining per persona after applying each filter. After applying all four filters, we are left with 31,850 ad images.

4.2.2 Identifying Targeted & Retargeted Ads

At this point, I do not know which of the ad images are retargets. Prior work has identified retargets by looking for specific URL parameters associated with them, however, this technique is only able to identify a subset of retargets served by DoubleClick [119]. Since my goal is to be mechanism- and platform-agnostic, I must use a more generalizable method to identify retargeted ads.

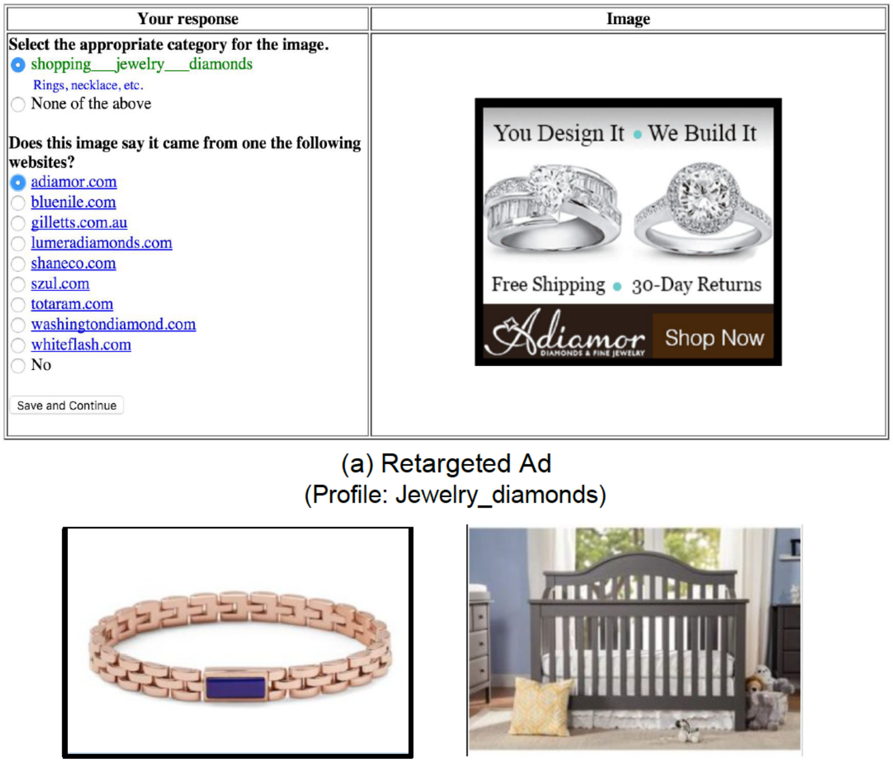
Crowdsourcing. Given a large number of ads in the corpus, I decided to crowdsource labels from workers on Amazon Mechanical Turk (AMT) [136]. I constructed Human Intelligence Tasks (HITs) that ask workers to label 30 ads, 27 of which are unlabeled, and 3 of which are known to be retargeted ads and serve as controls (I manually identified 1,016 retargets from our corpus of 31,850 to serve as these controls).

Figure 4.5(a) shows a screenshot of a HIT. On the right is an ad image, and on the left I ask the worker two questions:

1. Does the image belong to one of the following categories (with "None of the above" being one option)?
2. Does the image say it came from one of the following websites (with "No" being one option)?

The purpose of question (1) is to isolate behavioral and retargeted ads from contextual and untargeted ads (e.g., Figure 4.5(c), which was served to the *Music* persona). The list for question

[4] Several of our personas have retailers in common, which I account for when filtering ads.



Your response	Image
Select the appropriate category for the image. ● shopping___jewelry___diamonds Rings, necklace, etc. ○ None of the above Does this image say it came from one the following websites? ● adiamor.com ○ bluenile.com ○ gilletts.com.au ○ lumeradiamonds.com ○ shaneco.com ○ szul.com ○ totaram.com ○ washingtondiamond.com ○ whiteflash.com ○ No Save and Continue	

(a) Retargeted Ad
(Profile: Jewelry_diamonds)

(b) Behavioral Targeted Ad
(Profile: Jewelry)

(c) Normal Ad
(Profile: Music)

Figure 4.5: Screenshot of our AMT HIT, and examples of different types of ads.

(1) is populated with the shopping categories associated with the persona that crawled the ad. For example, as shown in Figure 4.5(a), the category list includes “shopping_jewelry_diamonds” for ads shown to our *Diamond Jewelry* persona. In most cases, this list contains exactly one entry, although there are rare cases where up to 3 categories are present in the list.

If the worker does not select “None” for question (1), then they are shown question (2). Question (2) is designed to separate retargets from behaviorally targeted ads. The list of websites for question (2) is populated with the e-commerce sites visited by the persona that crawled the ad. For example, in Figure 4.5(a), the ad clearly says “Adiamor”, and one of the sites visited by the persona is `adiamor.com`; thus, this image is likely to be a retargeted ad. Contrast this with Figure 4.5(b), which was served to our *Jewelry* persona, but does not include any text; in this case, it is unclear if the ad is a behavioral target or a retarget.

Quality Control. I apply four widely used techniques to maintain and validate the quality of our crowdsourced image labels [88, 177, 191]. *First*, I restrict our HITs to workers that have completed ≥50 HITs and have an approval rating of ≥95%. *Second*, I restrict our HITs to workers living in the US since our ads were collected from US websites. *Third*, I reject a HIT if the worker mislabels ≥2 of the control images (i.e., known retargeted ads); this prevents workers from being able to simply answer "None" to all questions. I resubmitted rejected HITs for completion by another worker. Overall, the workers correctly labeled 87% of the control images. *Fourth* and finally, I obtain **two labels** on each unlabeled image by different workers. For 92.4% of images, both labels match, so I accept them. I manually labeled the divergent images myself to break the tie.

Finding More Retargets. The workers from AMT successfully identified 1,359 retargeted ads. However, it is possible that they failed to identify some retargets, i.e., there are false negatives. This may occur in cases like Figure 4.5(b): it is not clear if this ad was served as a behavioral target based on the persona's interest in jewelry, or as a retarget for a specific jeweler.

To mitigate this issue, I manually examined all 7,563 images that were labeled as behavioral ads by the workers. In addition to the images themselves, I also looked at the inclusion chains for each image. In many cases, the URLs reveal that specific e-commerce sites visited by our personas hosted the images, indicating that the ads are retargeted. For example, Figure 4.5(b) is actually part of a retargeted ad from fossil.com. The manual analysis uncovered an additional 3,743 retargeted ads.

These results suggest that the number of false negatives from the crowdsourcing task could be dramatically reduced by showing the URLs associated with each ad image to the workers. However, note that adding this information to the HIT will change the dynamics of the task: false negatives may go down but the effort (and therefore the cost) of each HIT will go up. This stems from the additional time it will take each worker to review the ad URLs for relevant keywords.

In § 4.3.2, I compare the datasets labeled by the workers and by myself. Interestingly, although my dataset contains a greater *magnitude* of retargeted ads versus the worker's dataset, it does not improve *diversity*, i.e., the smaller dataset identifies 96% of the top 25 most frequent ad networks in the larger dataset. These networks are responsible for the vast majority of retargeted ads and inclusion chains in our dataset.

Final Results. Overall, I submitted 1,142 HITs to AMT. Workers were paid $0.18 per HIT, bringing the total cost of labeling to $415. I did not collect any personal information from workers. In total, I and the workers from AMT labeled 31,850 images, of which 7,563 are behaviorally targeted ads and 5,102 are retargeted ads. These retargets advertise 281 distinct e-commerce websites

(38% of all e-commerce sites).

4.2.3 Limitations

With any labeling task of this size and complexity, it is possible that there are false positives and negatives. Unfortunately, I cannot bound these quantities, since I do not have ground-truth information about known retargeted ad campaigns, nor is there a reliable mechanism to automatically detect retargets (e.g., based on special URL parameters, *etc.*).

In practice, the effect of false positives is that I will erroneously classify pairs of A&A domains as sharing information. I take measures to mitigate false positives by running a controlled crawl and removing images that appear in multiple personas (see § 4.2.1), but false positives can still occur. However, as I show in § 4.3, the results of my classifier are extremely consistent, suggesting that there are few false positives in our dataset.

False negatives have the opposite effect: I may miss pairs of A&A domains that are sharing information. Fortunately, the practical impact of false negatives is low, since I only need to correctly identify a single retargeted ad to infer that a given pair of A&A domains are sharing information. Given the size of our labeled dataset (5,102 retargets), it is likely that I have at least one retarget for all major pairs of collaborating A&A domains.

4.3 Analysis

In this section, I use the 5,102 retargeted ads uncovered in § 4.2, coupled with their associated inclusion chains (see § 4.1.2), to analyze the information flows between A&A domains. Specifically, I seek to answer two fundamental questions: *who* is sharing user data, and *how* does the sharing take place (e.g., client-side via cookie matching, or server-side)?

I begin by *categorizing* all of the retargeted ads and their associated inclusion chains into one of four classes, which correspond to different mechanisms for sharing user data. Next, I examine specific pairs of ad exchanges that share data and compare our detection approach to those used in prior works to identify cookie matching [5,65,119,147]. I find that prior work may be missing 31% of collaborating A&A domains. Finally, I construct a graph that captures A&A domains and the relationships between them and use it to reveal nuanced characteristics of the roles that different exchanges play in the ad ecosystem.

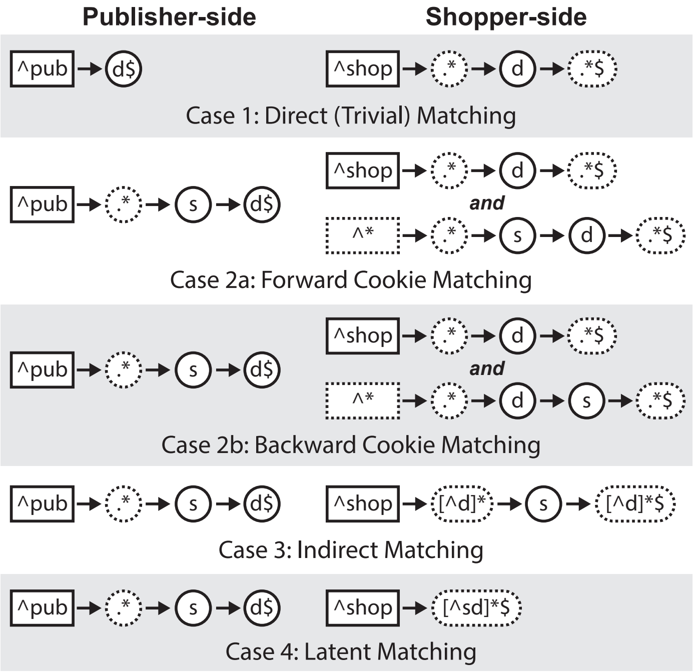


Figure 4.6: Regex-like rules we use to identify different types of ad exchange interactions. $shop$ and pub refer to chains that begin at an e-commerce site or publisher, respectively. d is the DSP that serves a retarget; s is the predecessor to d in the publisher-side chain, and is most likely an SSP holding an auction. Dot star $(.*)$ matches any domains zero or more times.

4.3.1 Information Flow Categorization

I begin the analysis by answering two basic questions: *for a given retargeted ad, was user information shared between A&A domains, and if so, how?* To answer these questions, I categorize the 35,448 *publisher-side* inclusion chains corresponding to the 5,102 retargeted ads in our data. Note that 1) we observe some retargeted ads multiple times, resulting in multiple chains, and 2) the chains for a given unique ad may not be identical.

I place publisher-side chains into one of the four categories, each of which corresponds to a specific information-sharing mechanism (or lack thereof). To determine the category of a given chain, I match it against carefully designed, regular expression-like rules. Figure 4.6 shows the pattern matching rules that I use to identify chains in each category. These rules are mutually exclusive, i.e., a chain will match one or none of them.

Terminology. Before I explain each classification in detail, I first introduce shared terminology that will be used throughout this section. Each retargeted ad was served to our persona via a *publisher-side* chain. pub is the domain of the publisher at the root of the chain, while d is the domain at the end of the chain that served the ad. Typically, d is a DSP. If the retarget was served via an auction, then an SSP s must immediately precede d in the publisher-side chain.

Each retarget advertises a particular e-commerce site. $shop$ is the domain of the e-commerce site corresponding to a particular retargeted ad. To categorize a given publisher-side chain, we must also consider the corresponding *shopper-side* chains rooted at $shop$.

4.3.1.1 Categorization Rules

Case 1: Direct Matches. The first chain type that I define are *direct matches*. Direct matches are the simplest type of chains that can be used to serve a retargeted ad. As shown in Figure 4.6, for us to categorize a publisher-side chain as a direct match, it must be exactly length two, with a direct resource inclusion request from pub to d. d receives any cookies they have stored on the persona inside this request, and thus it is trivial for d to identify our persona.

On the shopper-side, the only requirement is that d observed our persona browsing $shop$. If d does not observe our persona at $shop$, then d would not serve the persona a retargeted ad for $shop$. d is able to set a cookie on our persona, allowing d to re-identify the persona in the future.

I refer to direct matching chains as “trivial” because it is obvious how d is able to track our persona and serve a retargeted ad for $shop$. Furthermore, in these cases, no user information needs to be shared between A&A domains, since no ad auctions are being held on the publisher-side.

Case 2: Cookie Matching. The second chain type that I define are *cookie matches*. As the name implies, chains in this category correspond to the instance where an auction is held on the publisher-side, and we observe direct resource inclusion requests between the SSP and DSP, implying that they are matching cookies.

As shown in Figure 4.6, for us to categorize a publisher-side chain as cookie matching, s and d must be adjacent at the end of the chain. On the shopper-side, d must observe the persona at $shop$. Lastly, we must observe a request from s to d or d to s in some chain before the retargeted ad is served. These requests capture the moment when the two A&A domains match their cookies. Note that $s \rightarrow d$ or $d \rightarrow s$ can occur in a publisher- or shopper-side chain; in practice, it often occurs in a chain rooted at $shop$, thus fulfilling both requirements at once.

For this analysis, I distinguish between*forward* ($s \rightarrow d$) and *backward* ($d \rightarrow s$) cookie matches. Figure 2.2(a) shows an example of a forward cookie match. As we will see, many pairs of A&A domains engage in both forward and backward matching to maximize their opportunities for data sharing. To the best of my knowledge, no prior work examines the distinction between forward and backward cookie matching.

Case 3: Indirect Matching. The third chain type I define are *indirect matches*. Indirect matching occurs when an SSP sends meta-data about a user to a DSP, to help them determine if they should bid on an impression. With respect to retargeted ads, the SSP tells the DSPs about the browsing history of the user, thus enabling the DSPs to serve retargets for specific retailers, even if the DSP never directly observed the user browsing the retailer (hence the name, *indirect*). Note that no cookie matching is necessary in this case for DSPs to serve retargeted ads.

As shown in Figure 4.6, the crucial difference between cookie matching chains and indirect chains is that d *never* observes our persona at $shop$; only s observes our persona at $shop$. Thus, by inductive reasoning, we must conclude that s shares information about our persona with d, otherwise d would never serve the persona a retarget for $shop$.

Case 4: Latent Matching. The fourth and final chain type that I define are *latent matches*. As shown in Figure 4.6, the defining characteristic of latent chains is that neither s nor d observe our persona at $shop$. This begs the question: how do s and d know to serve a retargeted ad for $shop$ if they never observe our persona at $shop$? The most reasonable explanation is that some other ad exchange x that is present in the shopper-side chains shares this information with d behind-the-scenes.

I hypothesize that the simplest way for A&A domains to implement latent matching is by having x and d share the same unique identifiers for users. Although x and d are different domains and are thus prevented by the SOP from reading each others' cookies, both A&A domains may use the same deterministic algorithm for generating user IDs (e.g., by relying on IP addresses or browser fingerprints). However, as I will show, these synchronized identifiers are not necessarily visible from the client-side (i.e., the values of cookies set by x and d may be obfuscated), which prevents trivial identification of latent cookie matching.

Note: Although I do not expect to see cases 3 and 4, they can still occur. I explain in § 4.3.1.2 that indirect and latent matching is mostly performed by domains belonging to the same company. The remaining few instances of these cases are probably mislabeled behaviorally targeted ads.

4.3.1.2 Categorization Results

I applied the rules in Figure 4.6 to all 35,448 publisher-side chains in our dataset twice. First, I categorized the raw, unmodified chains; then I *clustered* domains that belong to the same companies,

Table 4.1: Results of categorizing publisher-side chains, before and after clustering domains.

	Unclustered		**Clustered**	
Type	**Chains**	**%**	**Chains**	**%**
Direct	1770	5%	8449	24%
Forward Cookie Match	24575	69%	25873	73%
Backward Cookie Match	19388	55%	24994	70%
Indirect Match	2492	7%	178	1%
Latent Match	5362	15%	343	1%
No Match	775	2%	183	1%

and categorized the chains again. For example, Google owns youtube.com, doubleclick.com, and 2mdn.net; in the clustered experiments, I replace all instances of these domains with google.com. § 8.1 lists all clustered domains.

Table 4.1 presents the results of our categorization. The first thing we observe is that cookie matching is the most frequent classification by a large margin. This conforms to our expectations, given that RTB is widespread in today's ad ecosystem, and major exchanges like DoubleClick support it [2]. Note that, for a given (s, d) pair in a publisher-side chain, we may observe $s \rightarrow d$ and $d \rightarrow s$ requests in our data, i.e., the pair engages in forward and backward cookie matching. This explains why the percentages in Table 4.1 do not add up to 100%.

The next interesting feature that we observe in Table 4.1 is that indirect and latent matches are relatively rare (7% and 15%, respectively). Again, this is expected, since these types of matching are more exotic and require a greater degree of collaboration between ad exchanges to implement. Furthermore, the percentage of indirect and latent matches drops to 1% when we cluster domains. To understand why this occurs, consider the following real-world example chains:

Publisher-side: $pub \rightarrow rubicon \rightarrow googlesyndication$

Shopper-side: $shop \rightarrow doubleclick$

According to the rules in Figure 4.6, this appears to be a latent match, since Rubicon and Google Syndication do not observe our persona on the shopper-side. However, after clustering the Google domains, this will be classified as cookie matching (assuming that there exists at least one other request from Rubicon to Google).

The above example is extremely common in our dataset: 731 indirect chains become cookie matching chains after we cluster the Google domains *alone*. Importantly, this finding provides strong evidence that Google does, in fact, use latent matching to share user tracking data between its various domains. Although this is allowed in Google's terms of service as of 2014 [79], my results provide direct evidence of this data sharing with respect to serving targeted ads. In the vast majority of these cases, Google Syndication is the DSP, suggesting that on the server-side, it ingests tracking data and user identifiers from all other Google services (e.g., DoubleClick and Google Tag Manager).

Of the remaining 1% of chains that are still classified as indirect or latent after clustering, the majority appear to be false positives. In most of these cases, we observe s and d doing cookie matching in other instances, and it seems unlikely that s and d would also utilize indirect and latent mechanisms. These ads are probably mislabeled behaviorally targeted ads.

The final takeaway from Table 4.1 is that the number of uncategorized chains that do not match any of our rules is extremely low (1-2%). These publisher-side chains are likely to be false positives, i.e., ads that are not actually retargeted. These results suggest that my image labeling approach is very robust since the vast majority of chains are properly classified as direct or cookie matches.

4.3.2 Cookie Matching

The results from the previous section confirm that cookie matching is ubiquitous on today's web and that this information sharing fuels highly targeted advertisements. Furthermore, my classification results demonstrate that we can detect cookie matching without relying on semantic information about cookie matching mechanisms.

In this section, I take a closer look at the pairs of A&A domains that we observe matching cookies. I seek to answer two questions: *first*, which pairs match most frequently, and what is the directionality of these relationships? *Second*, what fraction of cookie matching relationships will be missed by the heuristic detection approaches used by prior work [5, 65, 119, 147]?

Who Is Cookie Matching? Table 4.2 shows the top 25 most frequent pairs of A&A domains that we observe matching cookies. The arrows indicate the direction of matching (forward, backward, or both). "Ads" is the number of unique retargets served by the pair, while "Chains" is the total number of associated publisher-side chains. I present both quantities as observed in our complete dataset (containing 5,102 retargets), as well as the subset that was identified solely by the AMT workers (containing 1,359 retargets).

We observe that cookie matching frequency is heavily skewed towards several heavy-hitters. In aggregate, Google's domains are most common, which makes sense given that Google is the largest ad exchange on the web today. The second most common is Criteo; this result also makes sense, given that Criteo specializes in retargeted advertising [45]. These observations remain broadly true

Table 4.2: Top 25 cookie matching partners in my dataset. The arrow signifies whether we observe forward matches (→), backward matches (←), or both (↔). The heuristics for detecting cookie matching are: **DC** (match using DoubleClick URL parameters), **E** (string match for exact cookie values), **US** (URLs that include parameters like "usersync"), and - (no identifiable mechanisms). Note that the HTTP request formats used for forward and backward matches between a given pair of exchanges may vary.

			All Data		**AMT Only**		
Participant 1		**Participant 2**	**Chains**	**Ads**	**Chains**	**Ads**	**Heuristics**
criteo	↔	googlesyndication	9090	1887	1629	370	↔: US
criteo	↔	doubleclick	3610	1144	770	220	→: E, US ←: DC, US
criteo	↔	adnxs	3263	1066	511	174	↔: E, US
criteo	↔	googleadservices	2184	1030	448	214	→: E, US ←: US
criteo	↔	rubiconproject	1586	749	240	113	↔: E, US
criteo	↔	servedbyopenx	707	460	111	71	↔: US
mythings	↔	mythingsmedia	478	52	53	1	→: E, US ←: US
criteo	↔	pubmatic	363	246	64	37	→: E, US ←: US
doubleclick	↔	steelhousemedia	362	27	151	16	→: US ←: E, US
mathtag	↔	mediaforge	360	124	63	13	↔: E, US
netmng	↔	scene7	267	162	45	32	→: E ←: -
criteo	↔	casalemedia	200	119	54	31	→: E, US ←: US
doubleclick	↔	googlesyndication	195	81	101	62	↔: US
criteo	↔	clickfuse	126	99	14	13	↔: US
criteo	↔	bidswitch	112	78	25	15	→: E, US ←: US
googlesyndication	↔	adsrvr	107	29	102	24	↔: US
rubiconproject	↔	steelhousemedia	86	30	43	19	↔: E
amazon-adsystem	↔	ssl-images-amazon	98	33	33	7	-
googlesyndication	↔	steelhousemedia	47	22	36	16	-
adtechus	→	adacado	36	18	36	18	-
googlesyndication	↔	2mdn	40	19	39	18	→: US ←: -
atwola	→	adacado	32	6	28	5	-
adroll	↔	adnxs	31	8	26	7	-
googlesyndication	↔	adlegend	31	22	29	20	-
adnxs	↔	esm1	46	1	0	0	→: US ←: -

across the AMT and complete datasets: although the relative proportion of ads and chains from less-frequent exchange pairs differs somewhat between the two datasets, the heavy-hitters do not change. Furthermore, we also see that the vast majority of A&A pairs are identified in both datasets.

Interestingly, we observe a great deal of heterogeneity with respect to the directionality of cookie matching. Some boutique exchanges, like Adacado, only ingest cookies from other exchanges. Others, like Criteo, are omnivorous, sending or receiving data from any and all willing partners. These results suggest that some participants are more wary about releasing their user identifiers to other exchanges.

Comparison to Prior Work. I observe many of the same participants matching cookies as prior work, including DoubleClick, Rubicon, AppNexus, OpenX, MediaMath, and myThings [5,65,147]. Prior work identifies some additional ad exchanges that I do not (e.g., Turn); this is due to my exclusive focus on participants involved in retargeted advertising.

However, I also observe participants (e.g., Adacado and AdRoll) that prior work does not. This may be because prior work identifies cookie matching using heuristics to pick out specific features in HTTP requests [5, 65, 119, 147]. In contrast, my proposed categorization approach is content and mechanism agnostic.

To investigate the efficacy of heuristic detection methods, I applied three of them to my dataset. Specifically, for each pair (s, d) of exchanges that I categorize as cookie matching, I apply the following tests to the HTTP headers of requests between s and d or vice-versa:

1. I look for specific keys that are known to be used by DoubleClick and other Google domains for cookie matching (e.g., “google_nid” [147]).
2. I look for cases where unique cookie values set by one participant are included in requests sent to the other participant[5].
3. I look for keys with revealing names like “usersync” that frequently appear in requests between participants in our data.

As shown in the “Heuristics” column in Table 4.2, in the majority of cases, heuristics are able to identify cookie matching between the participants. Interestingly, we observe that the mechanisms used by some pairs (e.g., Criteo and DoubleClick) change depending on the directionality of the cookie match, revealing that the two sides have different cookie matching APIs.

However, for 31% of our cookie matching partners, the heuristics are unable to detect signs of cookie matching. I hypothesize that this is due to obfuscation techniques employed by specific ad exchanges. In total, there are 4.1% cookie matching chains that would be completely missed by heuristic tests. This finding highlights the limitations of prior work and bolsters the case for my content- and platform-agnostic classification methodology.

4.3.3 The Retargeting Ecosystem

In this last section, I take a step back and examine the broader ecosystem for retargeted ads that are revealed by our dataset. To facilitate this analysis, I construct a graph by taking the union of all of our publisher-side chains. In this graph, each node is a domain (either a publisher or an A&A), and edges correspond to resource inclusion relationships between the domains. My graph formulation differs from prior work in that edges denote actual information flows, as opposed to simple co-occurrences of trackers on a given domain [77].

[5] To reduce false positives, I only consider cookie values that have length $>$10 and $<$100.

Table 4.3: Overall statistics about the connectivity, position, and frequency of A&A domains in the dataset.

			Degree			Position p in Chains (%)			# of Shopper	# of
		Domain	**In**	**Out**	**In/Out Ratio**	p_2	p_{n-1}	p_n	**Websites**	**Ads**
DSPs		criteo	35	6	5.83	9.28	0.00	68.8	248	3,335
		mediaplex	8	2	4.00	0.00	85.7	0.07	20	14
		tellapart	6	1	6.00	25.0	100.0	0.18	33	9
		mathtag	12	6	2.00	0.00	90.9	0.06	314	2
		mythingsmedia	1	0	-	0.00	0.00	1.41	1	59
		steelhousemedia	8	0	-	0.00	0.00	16.8	40	89
		mediaforge	5	0	-	0.00	0.00	1.28	29	143
SSPs		pubmatic	5	9	0.56	3.17	74.2	0.01	362	4
		rubiconproject	19	22	0.86	23.5	62.8	0.01	394	3
		adnxs	18	20	0.90	94.2	91.9	0.16	476	12
		casalemedia	9	10	0.90	1.30	90.0	0.00	298	0
	AOL	atwola	4	19	0.21	84.6	18.2	0.01	62	2
		advertising	4	4	1.00	0.00	75.0	0.10	337	17
		adtechus	17	16	1.06	1.58	27.3	0.09	328	15
	OpenX	servedbyopenx	6	11	0.55	7.2	83.8	0.00	2	0
		openx	10	9	1.11	0.95	9.83	0.00	390	0
		openxenterprise	4	4	1.00	40.0	20.0	0.00	1	0
Google		googletagservices	44	2	22.00	93.7	0.00	0.00	65	0
		googleadservices	4	17	0.24	2.94	33.5	0.00	485	0
		2mdn	3	1	3.00	0.00	0.00	1.35	62	125
		googlesyndication	90	35	2.57	70.1	62.7	19.8	84	638
		doubleclick	38	36	1.06	38.8	63.1	0.22	675	19

Table 4.3 presents statistics on the top A&A domains in our dataset. The "Degree" column shows the in- and out-degree of nodes, while "Position" looks at the relative location of nodes within chains. p_2 is the second position in the chain, corresponding to the first ad network after the publisher; p_n is the DSP that serves the retarget in a chain of length n; p_{n-1} is the second to last position, corresponding to the final SSP before the DSP. Note that a domain may appear in a chain multiple times, so the sum of the p_i percentages maybe $>$100%. The last two columns count the number of unique e-commerce sites that embed resources from a given domain, and the unique number of ads served by the domain.

Based on the data in Table 4.3, we can roughly cluster the ad domains into two groups, corresponding to SSPs and DSPs. DSPs have low or zero out-degree since they often appear at position p_n, i.e., they serve an ad and terminate the chain. Criteo is the largest source of retargeted ads in our dataset by an order of magnitude. This is not surprising, given that Criteo was identified as the largest retargeter in the US and UK in 2014 [45].

In contrast, SSPs tend to have in/out-degree ratios closer to 1, since they facilitate the exchange of ads between multiple publishers, DSPs, and even other SSPs. Some SSPs, like Atwola, work more closely with publishers and thus appear more frequently at p_2, while others, like Mathtag, cater to other SSPs and thus appear almost exclusively at p_{n-1}. Most of the SSPs we observe also function as DSPs (i.e., they serve some retargeted ads), but there are "pure" SSPs like Casale

Media and OpenX that do not serve ads. Lastly, Table 4.3 reveals that SSPs tend to do more user tracking than DSPs, by getting embedded in more e-commerce sites (with Criteo being the notable exception).

Google is an interesting case study because its different domains have clearly delineated purposes. `googletagservices` is Google's in-house SSP, which funnels impressions directly from publishers to Google's DSPs: `2mdn`, `googlesyndication`, and `doubleclick`. In contrast, `googleadservices` is also an SSP, but it holds auctions with third-party participants (e.g., Criteo). `googlesyndication` and `doubleclick` function as both SSPs and DSPs, sometimes holding auctions, and sometimes winning auctions held by others to serve ads. Google Syndication is the second most frequent source of retargeted ads in our dataset behind Criteo.

Although we can develop heuristics like "Position in Chain" or "In/Out Degree Ratio" to characterize A&A domains, it is clear from Table 4.3 that determining the role of an A&A domain is not trivial. For example, I have categorized Atwola as an SSP since it appears at p_2 84.6% of the times, however, its in/out-degree ratio is 0.21; not close to 1 like known ad exchanges (e.g., DoubleClick). I discuss the challenges of categorizing A&A domains more in chapter 6. So, while this analysis gives us a rough idea of these roles, it will still contain several miscategorizations.

4.4 Summary

In this chapter, I developed a novel, principled methodology for detecting flows of tracking information between arbitrary A&A domains. This methodology is content- and platform-agnostic because it relies on the semantics of how exchanges serve ads, rather than focusing on specific cookie matching mechanisms. The key insight behind my approach is to leverage retargeted ads as a mechanism for identifying information flows. This is possible because the strict conditions that must be met for a retarget to be served allow us to infer the precise flow of tracking information that facilitated the serving of the ad.

Using an instrumented version of Chromium [17], I conducted extensive experiments using 90 *personas* to collect 35,448 inclusion chains associated with 5,102 unique retargeted ads (§ 4.1). Then, using regular-expression like matching rules (§ 4.3.1.1), these chains are categorized into four different categories. Through this categorization, I am able to reveal the underlying mechanism (i.e., cookie matching, server-side matching) used by A&A domains for information-sharing. As expected, cookie matching was the most common mechanism used to share user identifiers (§ 4.3.1.2).

Improvement Over Prior Work. My proposed methodology addresses the limitations of prior works [5,65,147] that rely on specific string patterns in HTTP content to detect information-sharing (cookie matching) among A&A domains. Since my methodology does not rely on HTTP content, it can detect both client- and server-side information-sharing flows. I demonstrate in § 4.3.1.2 that out of 200 cookie matching A&A partners I found, 31% of them would have been missed by heuristics used in prior works. Furthermore, I provided empirical evidence that Google shares tracking data across its services via server-side matching. Identification of such server-side information flows would not have been possible using techniques from prior works.

User's Digital Footprint. Data collected from these experiments can be crucial in understanding the true digital footprint of the user. That, in turn, can help develop effective privacy protection tools for users. For example, a privacy extension can inform users about the top x A&A domains that view user's information and can provide them an option to block them. In chapter 6, I use the data from this study to model an *accurate* picture of the user's privacy digital footprint.

Chapter 5

A Longitudinal Analysis of the `ads.txt` Standard

The primary goal of my dissertation is to study the privacy implications of Real Time Bidding (RTB) so that we can better understand the privacy digital footprint of the user in the modern ad ecosystem. In particular, I want to understand the information sharing between A&A domains which happens either to facilitate the RTB auctions via *cookie matching* or as a consequence of them. In chapter 4, I address the limitations of prior works by proposing a generic content- and platform-agnostic methodology to detect information sharing among arbitrary A&A domains. This solves one piece of the puzzle; we still need to understand how much privacy leakage happens through ad exchanges when they contact multiple DSPs during an ad auction (see § 1.1). To factor that into the user privacy model, we need an *accurate* list of A&A domains which act as ad exchanges. Identification of such a list is important since ad exchanges have this extra “power" to disperse tracking information to multiple ad networks during a single RTB auction. However, as I explained in § 4.3.3, identifying the roles of A&A domains in an automated way is not a trivial task.

Therefore, in this chapter, I make use of a recently introduced transparency standard called `ads.txt`. The `ads.txt` standard was introduced by the Interactive Advertising Bureau (IAB) in 2017 [167]. The main motivation behind the `ads.txt` standard is to tackle the issue of *domain spoofing*, which has long plagued the RTB ecosystem. However, as I will explain later, the data from this standard has the potential to identify a list of A&A domains that act as ad exchanges. So, I use the data from this standard as an opportunity to 1) isolate an *accurate* list od ad exchanges and 2) understand whether ad exchanges and DSPs are complying with the `ads.txt` standard in the effort to combat domain spoofing.

The complexity, scale, and opacity of the ad ecosystem create opportunities for various kinds of fraud. While *click* and *impression fraud* are longstanding problems [46, 49, 181, 183], RTB in particular has opened the door to a novel fraud known as *domain spoofing* [37, 98, 101]. In this attack, the fraudster creates fake bid requests for impressions that were purportedly generated by visitors to high-value *publishers* (e.g., CNN or YouTube). Advertisers/DSPs bid highly to show their ads on these valuable publishers, but the ads end up appearing on low-value websites, or nowhere at all, while the fraudster collects the profit. Attackers can earn millions of dollars per day spoofing bid requests [37].

The fundamental issue that enables domain spoofing is the opacity of the RTB ecosystem: advertisers cannot tell which auctioneers (exchanges) are *authorized* to sell impression inventory from a given publisher. This lack of transparency gives attackers the ability to spoof inventory from any publisher. To address this problem, the Interactive Advertising Bureau (IAB) Tech Lab introduced the `ads.txt` standard [167] in 2017. `ads.txt` is designed to rectify this transparency problem by allowing publishers to state, in a machine-readable format, which auctioneers are authorized to sell their impression inventory [83]. To opt-in to the standard, a publisher must place a file named `/ads.txt` at the root of their website; exchanges and DSPs can then download the file and verify the authenticity of bid requests.

In addition to helping mitigate domain spoofing, the `ads.txt` standard is of potential interest to researchers and privacy advocates. The opacity of the online advertising ecosystem has long frustrated attempts to understand which third-parties are part of the ecosystem, as well as the role of each third-party (e.g., tracker, advertiser, auctioneer, *etc.*). The practical consequence of this opacity is that users have grown suspicious of online advertisers and their privacy practices [11, 121, 187]. `ads.txt` fundamentally changes the landscape, by making it explicit which third-party domains in a given first-party context are *ad exchanges* (i.e., auctioneers). In aggregate, `ads.txt` data has the potential to reveal, for the first time, the relationships between publishers, ad exchanges, DSPs, and advertisers.

In this chapter, I take the first step towards measuring and quantifying the landscape revealed by `ads.txt`-compliant publishers. In particular, I aim to answer two basic questions:

1. *How useful is the* `ads.txt` *standard as a transparency mechanism?* This includes the scope, specificity, and correctness of the data contained in `ads.txt` files. This will potentially let me extract an *accurate* list of ad exchanges.

2. *Are members of the online ad ecosystem complying with the `ads.txt` standard?* This includes the adoption of the standard by publishers, as well as enforcement (or lack thereof) of the standard by ad exchanges and DSPs when bidding on impressions.

To answer these questions, I crawled `ads.txt` files from Alexa Top-100K websites every month between January 2018 and April 2019. I focus on these websites because their impressions are valuable, and thus they have the strongest incentive to adopt `ads.txt`. I also conducted monthly crawls of the Alexa Top-100K websites to gather information about the ad exchanges and other A&A domains that each website interacted with. This data allows me to observe whether auctioneers and DSPs appear to be in compliance with the rules stipulated in publishers' `ads.txt` files.

Through this study, I make the following key contributions and findings:

- I present the first large-scale, longitudinal study of `ads.txt`. I observe that as of April 2019, 20% of Alexa Top-100K websites have adopted the standard, which rises to 62% when we only consider websites that display ads via RTB auctions. This demonstrates that `ads.txt` has achieved impressive adoption since it was introduced in 2017.
- With respect to transparency, `ads.txt` allows us to identify 1,035 unique domains belonging to ad exchanges from 62% of the Alexa Top-100K publishers that display ads via RTB auctions. That said, I also find that `ads.txt` data has a variety of imperfections, and I develop methods to mitigate these deficiencies.
- With respect to compliance, I find that the vast majority of RTB ads in our sample were bought from authorized sellers. This suggests that ad exchanges and DSPs are complying with the standard. However, I also see that domain spoofing is still possible because major ad exchanges still accept impression inventory from publishers that have not adopted `ads.txt`. Further, I document cases where major ad exchanges purchased impressions from unauthorized sellers, in violation of the standard.

5.1 Background

In this section, I briefly introduce the rationale behind the `ads.txt` standard and discuss the standard in detail.

5.1.1 Ad Fraud and Spoofing

The online ad ecosystem has long been plagued with fraud, generating estimated losses of $8.2 billion per year in 2015 [95]. The most well-known forms are *impression fraud* and *click fraud* [49, 151, 181]. In this scheme, the attacker creates a seemingly-legitimate publisher and contracts with ad exchanges to sell their impressions. The attacker then earns revenue by directing fraudulent traffic to their own publisher. I discuss prior work on these forms of fraud in § 5.2.

The rise of programmatic advertising has created an opportunity for a different type of fraud known as *domain spoofing* or sometimes *inventory counterfeiting* [37, 98, 101]. In this scheme, the attacker generates bid requests that are supposedly for impressions on a high-value publisher (e.g., CNN or The New York Times), when in reality these impressions are either (1) entirely fabricated or (2) actually generated from a low-value publisher (which is often controlled by, or collaborates with, the attacker). Attackers can implement spoofing attacks by creating or compromising an SSP, or (in some cases) simply by setting up an illegitimate publisher. The attacker can make their spoofed inventory harder to detect by mixing it with legitimate inventory [183].

5.1.2 A Brief Intro to `ads.txt`

The fundamental flaw in the programmatic advertising ecosystem that enabled domain spoofing is that legitimate ad exchanges and DSPs had no way of knowing which ad exchanges/SSPs were *authorized* to sell impression inventory from a given publisher. This lack of transparency gave attackers the ability to spoof inventory from any publisher.

To combat spoofing, the Interactive Advertising Bureau (IAB) Tech Lab, which is a non-profit trade association for online advertisers, introduced the `ads.txt` standard [167]. The standard is designed to rectify the transparency issues that allowed spoofing to flourish, by allowing publishers to state, in a machine-readable format, which SSPs and ad exchanges are authorized to sell their impression inventory. To be compliant with the standard, ad exchanges and SSPs are supposed to not accept inventory they are not authorized to sell, while DSPs are not supposed to buy inventory from unauthorized sellers.

`ads.txt` 1.0 was introduced in May 2017 [167], and the latest 1.0.2 standard was published in March 2019 [83]. Google announced that by December 2018, DSPs in their exchange would purchase impressions that were authenticated via `ads.txt` by default [80, 89], i.e., a DSP would need to opt-out of the security measure if they wanted to purchase unauthenticated impressions. Google runs one of the largest ad exchanges [25], which created a strong incentive for publishers to

```
# CNN.com/ads.txt
google.com, pub-7439281311086140, DIRECT, f08c47fec0942fa0
rubiconproject.com, 11078, DIRECT, 0bfd66d529a55807
c.amazon-adsystem.com, 3159, DIRECT # banner, video
openx.com, 537153334, DIRECT # banner
openx.com, 540038342, DIRECT, a698e2ec38604c6 # banner
pubmatic.com, 156565, RESELLER, 5d62403b186f2ace # banner
pubmatic.com, 156599, DIRECT, 5d62403b186f2ace # banner
```

Listing 5.1: Example `ads.txt` taken from `cnn.com` on May 11, 2019 (and edited for brevity).

adopt `ads.txt` by the end of 2018 if they wanted their inventory to be purchasable by all DSPs in the auction.

5.1.3 `ads.txt` File Format

Much like the `robots.txt` exclusion standard [105], the `ads.txt` standard is instantiated by including a text file named `/ads.txt` at the root of a website. Listing 5.1 shows an example `ads.txt` file for illustrative purposes. `ads.txt` files obey a simple, line-oriented format; in keeping with the IAB specification [83], we refer to each line as a *record*. Each record contains three or four comma-separated fields that authorize a given SSP/ad exchange to sell impression inventory on behalf of the given publisher. The fields are:

1. **Seller Domain**: A domain name specifying the SSP or ad exchange that the publisher is authorizing to sell their impression inventory.
2. **Publisher ID**: A string that uniquely identifies the publisher's account within the ad system hosted by the company in field 1.
3. **Relationship**: Either "DIRECT" or "RESELLER" depending on whether the publisher is the contractual owner of the advertising account in field 2 (former) or that the publisher has contracted with a third-party to manage the account (latter).
4. **Certification Authority ID** (Optional): An ID that uniquely corresponds to the company in field 1. As of this writing, these IDs are assigned by the Trustworthy Accountability Group.[1]

Every <seller, publisher ID, relationship> triple uniquely defines a business relationship between the given seller and the publisher who authored the `ads.txt` file. Note that a given seller/publisher pair may have multiple business relationships, each encoded as a different record in the `ads.txt` file. As shown in Listing 5.1, this may happen if the publisher has multiple accounts with

[1] https://www.tagtoday.net/

the seller (field #2 varies) and/or because the publisher has DIRECT and RESELLER relationships with the seller (field #3 varies).

ads.txt files may also contain comments, delimited by the "#" character. These may appear on their own line or at the end of record lines. Further, ads.txt files may contain additional meta-data that appears in a "variable=value" format. In our dataset (described in § 5.3), I observe that this meta-data is rare, and I ignore it in this study.

The most confusing aspect of the ads.txt standard is that the seller domains listed in field #1 are not necessarily the domains that host ad auctions. For example, Google specifies that its seller domain is google.com, even though the actual auctions are hosted at doubleclick.net. Each SSP/ad exchange defines what domain should be placed in field #1 to authorize them.

5.2 Related Work

In this section, I survey the literature on the ecosystem of ad fraud and prevention mechanisms. I also discuss related work on the ads.txt standard.

5.2.1 Ad Fraud

Over the years, numerous white-papers and blog posts have been published by researchers and advertisers, documenting the issues pertaining to ad fraud. In 2016, the IAB published a white-paper highlighting that ad fraud costs advertisers $8.2B per year [95, 172]. Similarly, the Association of National Advertisers (ANA) reported ad fraud costs of $7.2B in 2016 [175]. Daswani et al. present an accessible introduction to the topic of ad fraud in [46].

Researchers have proposed methodologies to study various forms of ad fraud. Springborn et al. examined the extent of impression fraud by setting up honeypot websites [181]. Dave et al. provided a systematic look at click-spam and proposed an automated methodology to fingerprint click-spam attacks [49]. Some studies have provided case studies on botnets conducting click-spam [47, 131, 151]. Haddadi et al. [87] used bluff ads to detect click fraud. Stone-Gross et al. studied ad fraud in ad exchanges [183].

Several prevention mechanisms have also been introduced in the literature. Zhang et al. and Metwally et al. proposed methodologies to combat ad fraud by identifying duplicate clicks [129, 199]. Metwally et al. further proposed an approach to detect click fraud by looking for similarities among fraudsters [130]. Nazerzadeh et al. provided an approach based on economic incentives to counter ad fraud [139]. However, sophisticated botnets like *ZeroAccess* [173] and *ClickBot.A* [39] can evade such prevention mechanisms. Pearce et al. and Daswani et al. outlined techniques to combat fraud from botnets [47, 151]. WhiteOps published a report on their takedown of the infamous *Methbot* [128].

Domain spoofing has been a major issue in programmatic advertising. A good introduction to domain spoofing is provided in [98, 101]. Recently, *Methbot* spoofed domains for more than 6,000 premium publishers to generate revenue of $5M per day [37]. In November 2017, Adform published a white-paper describing how they took down *HyphBot*, which was generating 1.5B spoofed requests per day [94].

5.2.2 `ads.txt` Adoption

Besides a white-paper and some blog posts, to the best of my knowledge, there is no prior work that provides an in-depth, longitudinal analysis of the `ads.txt` standard.

Lukasz Olejnik, an independent researcher, recently published a white-paper on his longitudinal study of the `ads.txt` standard [145]. Olejnik gathered `ads.txt` data on Alexa Top-100K publishers from August 2017, right after the inception of the `ads.txt`, to March 2018. He performed one more crawl towards the end of December 2018. Results from this white-paper corroborate our findings regarding longitudinal trends in adoption and top sellers. Olejnik did **not** study the compliance aspect of the standard.

Since the inception of the `ads.txt` standard, several blog posts have studied its trends, and different companies have reported different trends. Pixalate reported a x5 growth in `ads.txt` adoption in 2018, with 75% of the top 1,000 programmatic domains adopting the standard [154]. They also claim that `ads.txt` has reduced ad fraud by 10% [155]. According to OpenX, 60% of the top 1,000 publishers (comScore's list) have adopted the standard [153]. First Impressions' reported adoption trends on Alexa Top-1000 sites are similar to ours [69]. Some blogs also noticed errors in publishers' `ads.txt` files [69, 153].

Several companies, including Google, provide tools for publishers to generate and validate their `ads.txt` records [9, 10, 80].

In their bid to eliminate the ability to profit from counterfeit inventory and bring more transparency to programmatic advertising, IAB has recently introduced a `ads.txt`-like standard for mobile apps, called app-`ads.txt` [84]. Furthermore, IAB is working towards introducing another standard called `sellers.json`, which will allow the buyers to discover the identities of all the

```
# Incorrect format, less than 3 comma separated fields
google.com - pub-7439281311086140, DIRECT
# Invalid seller domain, misspelled rubiconproject.com
rubicnproject.com, 17380, DIRECT, 0bfd66d529a55807
# doubleclick.net is incorrect, should be google.com
doubleclick.net, pub-7439281311086140, DIRECT
```

Listing 5.2: Example `ads.txt` containing different classes of errors in each record.

authorized reseller partners of a participating seller (SSP) [85].

5.3 Methodology

The goal of this study is to monitor publishers' adoption of the standard, the involvement of authorized sellers (exchanges/SSPs), and compliance with the standard by buyers (DSPs). In this section, I outline how I collected and cleaned `ads.txt` data. Then I describe how I collected resource inclusions from publishers to determine compliance with the `ads.txt` standard.

5.3.1 Collection of `ads.txt` Data

The most crucial dataset for our study is `ads.txt` files from publishers. To obtain this data, I started crawling the Alexa Top-100K websites on January 15, 2018. Up until December 1, 2018, I repeated the `ads.txt` crawl every 15 days. After that, I crawled once every 30 days (on the 1^{st} of each month). The latest snapshot used in this study is from April 1, 2019. Overall, I performed 26 crawls.

After the start of our data collection, Scheitle et al. [171] and others [158, 170] published compelling analyses that document instabilities in the Alexa ranking. Considering these results, from October 15, 2018 onwards, I started updating the list of target websites in my crawl: before each crawl, I fetched the latest Alexa Top-100K list, computed the union of it and my existing list of target websites, and crawled the result. Subsequently, my sample size grew from 100K websites on January 15, 2018, to 240K on April 1, 2019.

According to the IAB standard, the `ads.txt` file must be placed at the root of a given domain. I used Python's `requests` module to fetch the `ads.txt` files: for each publisher p from the Alexa Top-100K, I accessed the `/ads.txt` URL from p's root. I sent a valid `User-Agent` with each request. I was able to crawl all the target websites within 2–3 hours by parallelizing across a 16-node cluster at *Northeastern University*.

5.3.1.1 Parsing and Cleaning

To facilitate analysis, I parsed all of the `ads.txt` files gathered by the crawler. In theory, `ads.txt` files are supposed to obey the IAB specified format outlined in § 5.1.3; in practice, I observed many files with errors, which necessitated that I develop a custom approach for parsing and validating `ads.txt` files.

I observed that publishers made a variety of mistakes in their `ads.txt` files, of which I highlight three examples in Listing 5.2. Some records, such as the first in Listing 5.2, contain syntactic errors, i.e., they do not obey the formatting specification. Other records contained semantic errors. For example, the second record in Listing 5.2 is in the correct format, but the seller is incorrect: it is supposed to be `rubiconproject.com`, but is `rubicnproject.com` instead. The third record in Listing 5.2 illustrates an even subtler error, where the seller domain has been accidentally replaced by a related, but incorrect, domain. In this case, the seller should be `google.com` but was mistakenly added as `doubleclick.net`.

I used a multi-stage filtering process to remove records with syntax errors and some semantic errors. First, I discarded all records that did not conform to the `ads.txt` specification (e.g., the first record in Listing 5.2). Second, I extracted all 2,381 unique seller domains S from the syntactically valid records in our dataset. Third, to identify semantically invalid domains (like the second record in Listing 5.2), I queried each domain in the WHOIS database. I was able to find WHOIS data for 1,035 of the seller domains. To make sure that I did not have any false negatives (i.e., the WHOIS crawl failed to fetch data for a valid seller domain), I also performed DNS resolution on all the negative samples. None of the domains in the negative sample had a successful resolution. Therefore, unless mentioned otherwise, I only consider the 1,035 seller domains S_v in my analysis. Further, I disregard all records containing the 1,346 unresolvable seller domains.

My filtering method cannot identify semantic errors like in the third record in Listing 5.2 because, in these cases, the erroneous domains are valid and resolvable. As I discuss in § 5.4.2, I estimate that ~20% of the unique sellers in my dataset are the result of such errors, but these low-frequency sellers end up having very limited impact on the analysis.

5.3.1.2 Collecting Resource Inclusions

To assess compliance with the `ads.txt` standard on an `ads.txt`-enabled publisher, I need to examine which sellers and buyers were involved in serving ads through RTB auctions. To accomplish this, I rely on *inclusion trees* described in § 4.1.2[2].

Using this tool, I repeatedly drove a Chrome browser to collect resource inclusions for all the publishers from the ads.txt crawl. These crawls were done right after each ads.txt crawl finished (see § 5.3.1). In particular, for each publisher p in the dataset, the crawler visited the homepage for p, then iteratively crawled 15 randomly selected links that pointed to p. During these crawls, I presented a valid User-Agent, scrolled pages to the bottom, and waited for $\sim$10 seconds between subsequent page visits.

Once I had collected inclusion trees from publishers, I decomposed them into *inclusion chains* to facilitate analysis. For a given inclusion tree (corresponding to a single visit of a webpage), the chains are simply all of the root-to-leaf paths in the tree.

5.3.1.3 Detecting Ads

The last step in the methodology is identifying all of the inclusion chains that correspond to the serving of an ad. I do this by applying a series of filters: first, I eliminate all chains where the final resource is not an image. Second, I filter out chains where the final image is $\leq 50\times50$ pixels.[3] Finally, I filter out chains that include zero requests to a URL that matches a rule in EasyList [54]. This last step allows me to separate benign images from advertisements by ensuring that a known advertising-related URL was involved in serving the image.

5.4 Adoption of **ads.txt**

In this section, I analyze the adoption of the ads.txt standard over our 15-month study. I examine adoption trends from the perspective of Alexa Top-100K publishers and top sellers that appear in the ads.txt files.

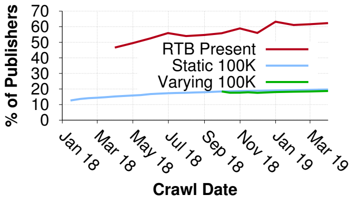


Figure 5.1: Adoption of the ads.txt standard by Alexa Top-100K publishers over time.

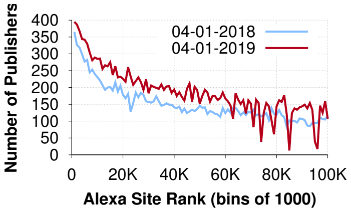


Figure 5.2: Publisher adoption over alexa ranks.

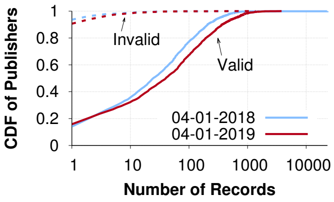


Figure 5.3: Number of ads.txt records per publisher.

5.4.1 Publisher's Perspective

I begin by examining the ads.txt standard from the perspective of publishers, starting with the adoption of the standard by Alexa Top-100K websites over time. The *Static 100K* line in Figure 5.1 shows adoption by a static set of Alexa Top-100K websites that was sampled in January 2018. The Varying 100K line shows adoption by a dynamic set of Alexa Top-100K websites that grows over time to incorporate newly popular sites (see § 5.3.1). In January 2018, we observed 12.7% of websites adopting the standard, which grew steadily to 19.7% in April 2019. Adding new, popular websites over time had a negligible impact on our results. Further, my observations match those of Lukasz Olejnik, an independent researcher who has also been tracking ads.txt adoption [145].

Although adoption of ads.txt by Alexa Top-100K websites is modest overall, this baseline is too liberal since it includes websites that (1) do not display ads or (2) do not display ads via ad exchanges (e.g., Facebook, YouTube). There is no reason for these classes of websites to adopt ads.txt. To account for this, I isolate the set of websites W_{RTB} from our complete set of crawled websites W that appear to be displaying ads via RTB auctions. At a high-level, website $w \in W$ is also a member of W_{RTB} if we observe $\geq$1 inclusion chain rooted at w that includes $\geq$1 requests to a known ad exchange. I derive this list of known ad exchanges from the ads.txt data itself; see § 5.5.2 for further details.

The *RTB Present* line in Figure 5.1 shows adoption of ads.txt over time by websites in

[2] By the time of this study, Google had release Chrome Debugging Protocol (CDP) [38], which grants fine-grained access to Chrome's internals without the need to instrument the browser source code. We modified our tool to collect inclusion trees using CDP. In terms of inclusions, we still get the same information; CDP just provides more fine-grained information. To capture dynamic inclusions, scriptParsed events were used in the Debugger domain, and requestWillBeSent and responseReceived events were used in the Network domain. Through scriptParsed, JavaScript triggered by remote and inline scripts was tracked, whereas requestWillBeSent and responseReceived were used to observe any further resource requests. iframe inclusions were captured by collecting frameNavigated events in the Page domain.

[3]These images are too small to be ads; most are 1×1 tracking pixels. We chose 50×50 since it is smaller than any of the typical online advertising format [42, 43].

W_{RTB}. We observe that adoption has increased from 46.6% to 62.3% over the 15-months of this study[4]. Thus, although the majority of popular, ad-revenue supported publishers on the web have adopted ads.txt, there is still a significant number that remain vulnerable to ad inventory fraud attacks (see § 5.1.1).

Alexa Rank of ads.txt Publishers. Next, I investigate how ads.txt adoption varies by publisher popularity. Figure 5.2 shows the frequency count of publishers with ads.txt files binned into groups of 1,000 by Alexa rank, drawn from two snapshots taken one year apart. Although adoption is uniformly higher in April 2019 as compared to April 2018, across both snapshots we see the same trend: publishers with high Alexa ranks have higher ads.txt adoption. For example, the adoption rate is ∼40% for Alexa Top-1K publishers. This is a positive, if somewhat expected trend, since popular (i.e., lucrative) publishers may be higher-value targets for ad inventory fraud attacks.

5.4.1.1 Correctness

Now that I have identified all publishers with ads.txt files in each snapshot, I can start analyzing the contents of these files. For a given publisher p, I validate all the records in its ads.txt file according to the IAB specification to identify syntactic errors (see § 5.1.3). Note that at this point, I do not attempt to validate the correctness of sellers; I defer this analysis to § 5.4.2.

Figure 5.3 shows the number of valid and invalid records in ads.txt files for all the publishers in two snapshots. Our first observation is that the size of ads.txt files grew between April 2018 and 2019: the number of valid records increased from 25 to 40 at the 50^{th} percentile over this year.[5] This occurred because existing publishers added more sellers to their files and because new publishers with relatively long ads.txt files adopted the standard over the year-long period. Our second observation is that a minority of publishers have large ads.txt files: 33% of publishers have ads.txt files with ≥100 valid entries, and 1% have ≥1000 valid entries. Broadly speaking, there are two types of websites that fall into these ranges: (1) well-known publishers like cnn.com and espn.com that have a large, valuable impression inventory and thus maintain relationships with many ad exchanges, or (2) platforms like wordpress.com and ucoz.com that provide hosting for thousands of small, independent publishers. Our final observation from Figure 5.3 is that 10% of the publishers have ≥1 invalid record in their ads.txt file.

[4] The inclusion crawls failed to tag image resources for the first 3 snapshots. That is why RTB Present line in Figure 5.1 starts from April 2018.

[5] This observation also matches Lukasz Olejnik's findings [145].

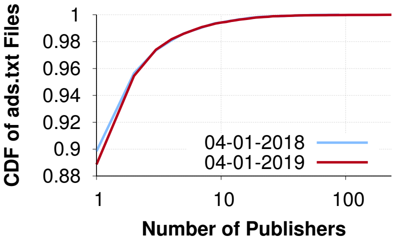


Figure 5.4: Number of publisher using the same `ads.txt` file.

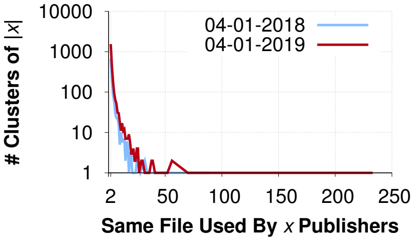


Figure 5.5: Clusters of $|x|$, where x is the # of publishers using the same file.

Table 5.1: Top 10 clusters of publishers using the same `ads.txt` file.

	Cluster	Unique Whois		Unique Whois		Unique Whois			# IPs	
#	Size	Servers	(Empty)	Registrars	(Empty)	Emails	(Empty)	Comments	/24	/16
1	233	19	(1)	19	(1)	12	(53)	Redirected to `ads.adthrive.com/sites/UNIQ_ID/ads.txt`.	156	71
2	198	23	(3)	25	(0)	13	(51)	Use `ads.txt` provided by `MediaVine`.	155	73
3	178	1	(177)	2	(176)	1	(177)	Sub-domains of `livejournal.com`, and use it's `ads.txt`.	2	2
4	106	1	(0)	1	(0)	1	(0)	Redirected to `ads.iacapps.com/generic/ads.txt` by `MindSpark Interactive`.	2	2
5	97	1	(0)	1	(0)	1	(0)	All owned by `Vox Media`.	7	1
6	73	6	(1)	8	(0)	4	(37)	Same website publishing platform used.	28	6
7	70	2	(68)	2	(68)	5	(6)	Sub-domains of `uol.com.br`.	11	7
8	56	4	(46)	12	(24)	8	(37)	Same website format (search engine). Mostly linking to `izito.*` and `zapmeta.*`.	5	4
9	56	1	(0)	1	(0)	1	(0)	Same website format (news). Same registrar and corresponding email.	4	4
10	52	16	(9)	19	(6)	16	(16)	All domains provide free video streaming (mostly for movies and porn).	48	25

5.4.1.2 Clustering Publishers Using `ads.txt`

In theory, each publisher should have a unique `ads.txt` file, since they have unique IDs in each exchange marketplace (see § 5.1.3). However, I observed some publishers distributing identical `ads.txt` files.

To investigate this surprising finding I plot Figure 5.4, which shows the number of publishers distributing each unique `ads.txt` file in our dataset. I find that $\sim$ 10% of the `ads.txt` files are distributed by $>$1 publisher and that this fraction is invariant over time. The most common `ads.txt` file in our dataset was distributed by 233 publishers in the April 2019 snapshot. Figure 5.5 shows the number of clusters of size x, where a cluster is defined as a group of publishers distributing the same `ads.txt` file. For example, there is a single cluster of publishers of size 233, and 1,539 clusters of size two distributing identical files.

To gain a better understanding of why these publishers are distributing identical `ads.txt` files, I manually analyzed the top 10 largest clusters. For each cluster, I (1) crawled the WHOIS registry data for its constituent publishers and (2) resolved the publisher domains to IP addresses and checked how many belonged to the same /24 and /16 subnets. Additionally, I randomly sampled 20 websites from each cluster and manually inspected their homepages and `ads.txt` files.

The results of this investigation are shown in Table 5.1. For each of the top-10 clusters, I show the number of unique servers, registrars, and contact email addresses from WHOIS associated with

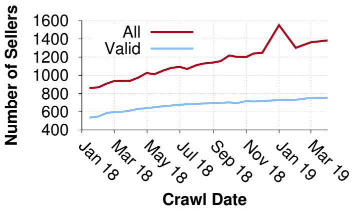


Figure 5.6: Number of seller domains over time.

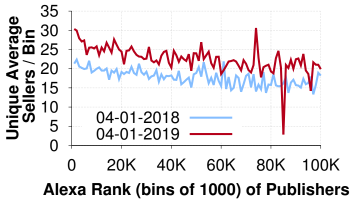


Figure 5.7: Authorized sellers over Alexa.

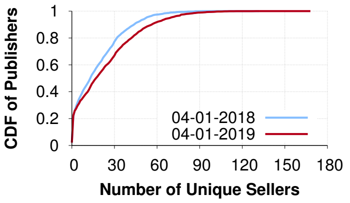


Figure 5.8: Sellers across two snapshots.

publishers in that cluster, as well as the number of unique /16 and /24 IP address ranges containing the publisher's IP addresses. For most of the clusters, the WHOIS information was shared across most or all of the individual clusters, strongly suggesting that the publishers in the cluster share a common owner or at least common management. The exceptions are clusters #3, #7, and #8, where most of the WHOIS records were private (and thus labeled as "empty" in our dataset). I see similar overlap with respect to IP address prefixes for clusters #3–5, #8, and #9, which is suggestive of common hosting infrastructure.

Manual investigation revealed three reasons for these large clusters of publishers. First, several clusters represent media properties with a common owner. For example, all of the publishers in cluster #5 were owned by Vox Media. Clusters #4, #8, #9, and #10 also each appear to have a single owner, respectively. Second, several clusters represented media platforms that host independent publishers, including clusters #3 (LiveJournal) and #7 (UOL). Third, several clusters represent independent publishers that happen to use consolidated SSP services. In particular, AdThrive (cluster #1) and MediaVine (#2) both appear to use their own publisher IDs when selling impression inventory, rather than having their pool of publishers all sign up for individual accounts with the ad exchanges.

5.4.2 Seller's Perspective

In this section, I shift perspective to focus on the sellers that are listed in ads.txt files. Sellers are the most important part of an ads.txt file since the whole point of the standard is for publishers to authorize sellers to sell their inventory.

To perform this analysis, I must first filter out the erroneous sellers that appear in ads.txt files. As described in § 5.3.1.1, I leverage WHOIS registry data and DNS resolution to identify all the syntactically invalid seller domains. Figure 5.6 shows the number of unique sellers I observe in each crawl before (*All* line) and after (*Valid* line) I filter out invalid sellers. I observe that the total

Table 5.2: Top 20 publishers with most sellers. Direct and Reseller are their seller account relationships.

	Alexa	# Unique	Valid	Relationship	
Publisher	Rank	Sellers	Entries	D	R
arcamax.com	22565	168	3617	434	3183
breitbart.com	242	158	980	123	857
walterfootball.com	48279	148	2805	394	2411
investing.com	408	130	1551	218	1333
webconsultas.com	13730	127	2309	263	2046
shoppinglifestyle.com	72547	119	1249	155	1094
moretvtime.com	17380	118	2408	231	2177
newindianexpress.com	13028	118	1967	225	1742
americanlisted.com	53358	117	1239	146	1093
thehindu.com	1067	117	1210	127	1083
thegatewaypundit.com	8429	116	1501	217	1284
vikatan.com	6005	114	1046	168	878
flvto.biz	889	114	3490	289	3201
realgm.com	11118	112	1397	186	1211
fayerwayer.com	18578	111	1944	12	1932
publimetro.co	40324	111	1944	12	1932
pjmedia.com	16437	111	1522	140	1382
metroecuador.com.ec	27378	111	1944	12	1932
nuevamujer.com	40645	111	1944	12	1932
publimetro.com.mx	21623	111	1944	12	1932

Table 5.3: Top 20 sellers with presence on most publishers. Publishers have either Direct, Reseller, or Both relationships with them.

	# of	Relationship			Avg. (Median)
Authorized Seller	Publishers	D	R	B	Entries / Publisher
google.com	17771	5305	1408	11058	14.39 (4.00)
appnexus.com	12825	578	5127	7120	15.24 (8.00)
rubiconproject.com	12691	1145	4969	6577	8.35 (5.00)
openx.com	12250	652	5432	6166	13.04 (7.00)
pubmatic.com	12112	605	6345	5162	13.80 (7.00)
indexexchange.com	11347	977	4713	5657	6.22 (4.00)
contextweb.com	10405	275	7214	2916	7.97 (4.00)
spotxchange.com	10197	292	7046	2859	7.16 (4.00)
spotx.tv	9957	299	7009	2649	6.64 (4.00)
advertising.com	9819	310	6705	2804	7.48 (4.00)
sovrn.com	9146	1612	3925	3609	3.97 (2.00)
adtech.com	9110	1103	4803	3204	4.61 (3.00)
freewheel.tv	9029	170	6729	2130	23.52 (7.00)
tremorhub.com	8529	260	6955	1314	5.32 (3.00)
smartadserver.com	8401	441	5836	2124	5.67 (3.00)
districtm.io	7599	1730	2015	3854	3.23 (2.00)
lkqd.net	7300	54	5589	1657	4.78 (3.00)
aolcloud.net	7298	855	4732	1711	3.31 (2.00)
lijit.com	7100	2236	2210	2654	3.11 (2.00)
teads.tv	6757	3406	1976	1375	2.49 (2.00)

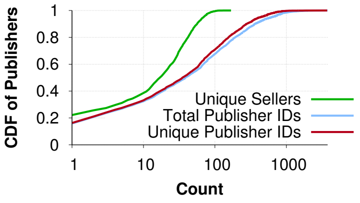


Figure 5.9: Number of sellers and associated publisher IDs (April 2019).

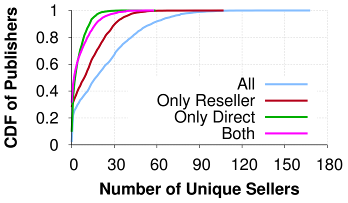


Figure 5.10: Sellers by publisher relationships (April 2019).

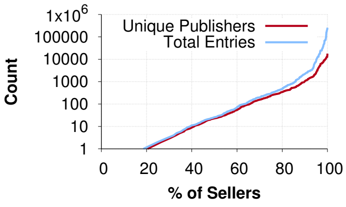


Figure 5.11: Number of unique publishers and total entries (across all publishers) for sellers.

number of sellers increases from 860 to 1,400 overtime, with the union over time containing 2,381 sellers. However, after I filter out the invalid sellers, the number of seller domains grows at a modest rate. This result is expected since it requires significant effort for new SSPs and ad exchanges to establish themselves in the marketplace.

The union of valid sellers overtime is 1,035 unique sellers, i.e., 56.4% of the seller domains in the ads.txt files contained syntactic errors. I focus on these sellers for the remainder of our analysis. Note that this set **over-estimates** the number of valid sellers, since it may include semantically incorrect sellers. Figure 5.11 (discussed later) indicates that up to 20% of the unique sellers may be erroneous due to semantic errors, however, these sellers only appear in a single ads.txt file throughout our dataset, meaning they have very limited impact on our analysis.

Sellers Per Publisher. Next, I compare the Alexa rank of publishers versus the number of sellers they authorize in their ads.txt files. Figure 5.7 presents the average number of valid sellers across bins of 1000 publishers sorted by their Alexa rank, with separate lines for our April 2018 and 2019 snapshots. We see that the average number of sellers at every rank has grown over a year: there were $\sim$ 10 more sellers per bin in the April 2019 snapshot as compared to April 2018. This is primarily due to publishers forming new partnerships with existing sellers, rather than the emergence of new sellers over time (see Figure 5.6). Additionally, I find that publishers at higher ranks have listed more authorized sellers on average, possibly because their impression inventory is more valuable, thus making them more desirable partners to ad exchanges.

Figure 5.8 shows the number of unique sellers listed within each publisher's ads.txt file for two snapshots of our crawl. I make three observations: first, ~2% of the publishers have no sellers in their files. I manually examined these ads.txt files and found that they were either empty or just contained comments (e.g., https://www.youtube.com/ads.txt). These empty ads.txt files are intentionally installed by publishers since they signal to ad exchanges and DSPs that **nobody** is authorized to sell their impressions. Second, the median publisher listed 17 sellers in their ads.txt, while the top 20% of publishers listed ≥42 unique sellers in their ads.txt files. Finally, we see that the number of unique sellers per publisher has increased slightly year-over-year, with the increases mostly concentrated amongst the publishers with the largest ads.txt files.

Table 5.2 focuses on the top 20 publishers who have listed the most unique sellers in their ads.txt files.[6] One interesting observation is that there is no correlation between Alexa rank and unique sellers for the top 20 publishers. They do have a common theme though — they are all news websites. Another notable observation is the difference between the number of unique sellers and the number of valid entries per publisher. The latter is an order of magnitude greater than the former because a publisher can have multiple publisher IDs associated with a given seller (see § 5.1.3). This is highlighted in Figure 5.9, which compares the count of unique sellers, total publisher IDs, and unique publisher IDs per publisher for ads.txt files in our April 2019 snapshot. We see an order of magnitude more publisher IDs than unique sellers. This conclusion remains the same even if I de-duplicate publisher IDs, which makes sense because duplicate publisher IDs within a given ads.txt file would be errors.

Recall that each publisher ID associated with a seller also has a specific relationship with the seller. This relationship can be of two types: *Direct* or *Reseller* (see § 5.1.3). For example, as shown in Table 5.2, arcmax.com has 3,617 publisher IDs for 168 unique sellers. Out of these 3,617 IDs, 434 have a *Direct* relationship, meaning the publisher directly controls the given account. For the

[6] Others have also observed that sites like arcamax.com and breitbart.com have unusually large ads.txt files [145, 184].

remaining 3,183 *Reseller* IDs, the publisher has authorized another entity to control this account associated with the seller.

Figure 5.10 breaks down the valid entries in each publishers' ads.txt files by relationship type for our April 2019 snapshot. The *All* line is identical to Figure 5.8, and is shown here for scale. The *Only* lines count cases where a publisher only has a *Direct* or *Reseller* relationship with a seller, while the *Both* line counts cases where the publisher has both relationships with a given seller. Overall, we see that *Reseller* relationships are most common: 25% of the publishers have only *Reseller* relationships with ≥20 sellers, whereas just 2% of the publishers have only *Direct* relationship with ≥20 sellers. The *Both* line is almost coincident with the *Only Direct* line, suggesting that when a publisher has a *Direct* relationship with a seller, they almost always have a *Reseller* relationship with that seller as well.

Seller Popularity. So far, I have looked at authorized sellers with respect to each publisher. Now, I look at the popularity of sellers across all publishers in our dataset.

Figure 5.11 shows each sellers' popularity in terms of (1) the total number of entries they appear in across all publishers, and (2) the number of unique publishers they have relationships with. I observe that 20% of the sellers are only involved with a single publisher. Some of these sellers are semantic errors (e.g., googlesyndication.com instead of google.com), some are typos (e.g., comgoogle.com), and some are legitimate ad *networks* (not exchanges, e.g., zergnet.com) that have been added to the ads.txt file by mistake (see § 5.3.1.1). At the other extreme, the top 25% and top 10% of sellers are listed on ≥250 and ≥1050 publishers, respectively. This result is expected since there are powerful network effects that draw publishers to the biggest ad exchange markets. Lastly, the top sellers have an order of magnitude more entries in comparison to their publisher presence. This bolsters our finding that publishers tend to register multiple accounts with top sellers.

Table 5.3 shows the top 20 sellers listed in the ads.txt files in our dataset. Unsurprisingly, the top ad companies like Google, OpenX, and Rubicon are present in this list. google.com is the most popular seller and is associated with 17.7K publishers. Furthermore, it appears in 14.4 entries per ads.txt file on average. From the table, we can see that publishers tend to have both direct and reseller relationships with the top sellers.

5.5 Compliance

In § 5.4, I looked at how Alexa Top-100K publishers have adopted the ads.txt standard over the course of 15-months, and which ad sellers they have authorized to sell their inventory during RTB auctions. In this section, I take the next step and try to examine the ads.txt standard from the ad buyers' side. After all, one of the major goals of ads.txt is to enable ad buyers (e.g., DSPs) to verify the authenticity of inventory before bidding. Thus, I pose the question: *are buyers complying with the ads.txt standard by only purchasing impression inventory via authorized sellers?*

5.5.1 Isolating RTB Ads

To determine whether ad buyers are complying with the ads.txt file for a given publisher p, I first need to identify ads that were served through RTB auctions on p. This is important since ads.txt compliance should only matter for RTB auctions.

Using the methodology from § 5.3.1.2, I extract all inclusion chains rooted in p. Then, as described in § 5.3.1.3, I use EasyList to identify all chains that eventually serve an ad on p. From these *ad inclusion chains*, I can further isolate just the ads served via RTB using two insights. First, I know that for an ad to be served via RTB, there must be at least 3 parties involved: the publisher, the exchange (seller), and the DSP (buyer). Thus, I filter out all the ad inclusion chains with $<$ 3 resources. Second, through the ads.txt dataset, I have a lower-bound estimate on all the ad exchanges (sellers) used by Alexa Top-100K publishers (set S_v, see § 5.3.1.1). Using these 1,035 sellers, I filter out all ad inclusion chains that have zero resources from the set of valid sellers.

After applying all the filters above, I am left with 135M RTB ad inclusion chains. Although I cannot claim that these chains capture all of the ads in our dataset served by RTB, they should cover the ads served by authorized sellers listed in ads.txt files.

5.5.2 Compliance Verification Metrics

Now that I have isolated the inclusion chains that served RTB ads, I can investigate compliance with the ads.txt standard by ad buyers. To this end, I create a set R_p of seller–buyer tuples (s, b) for each publisher p. s and b are derived from RTB ad inclusion chains, such that s and b are the 2^{nd}-level domains of the chain elements at index i and $i+1$ respectively. For example, consider an ad inclusion chain $p \rightarrow e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow d$, rooted at publisher p. The last element of the chain

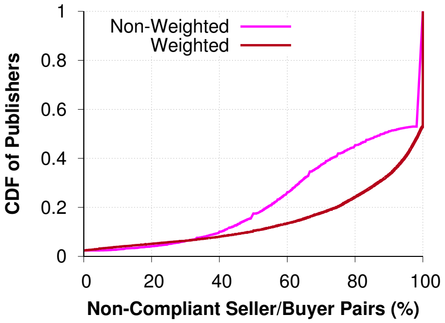


Figure 5.12: Percentage of non-compliant (non-clustered) Seller/Buyer tuples per publisher.

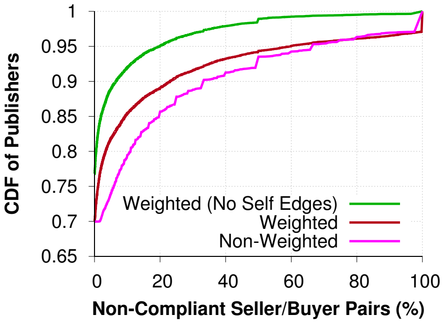


Figure 5.13: Percentage of non-compliant (clustered) Seller/Buyer tuples per publisher.

d is the DSP that ultimately served the ad. e_1, e_2 are both exchanges, and are present in the set of valid authorized sellers S_v, whereas $e_3 \notin S_v$. In this case, I would produce the buyer–seller tuples (e_1, e_2) and (e_2, e_3), since e_2 bought and then resold the impression. Lastly, note that since I only include tuples where s is a member of the ads.txt authorized sellers set S_v, I do not consider the tuple (e_3, d) in R_p.

I derive the set of non-compliant (s, b) tuples $R_p^\star$ for p, such that $s \notin S_p$, where S_p is the set of authorized sellers listed by p in its ads.txt file. Intuitively, the tuples in $R_p^\star$ capture cases where a seller was not authorized by the publisher to sell its inventory. Using $R_p^\star$, I calculate *unweighted compliance* for p as the percentage of non-compliant tuples over the total tuples $100 * |R_p^\star|/|R_p|$. However, this metric is not necessarily fair, since it does not take into account the relative frequency that sellers–buyer pairs appear in the ad inclusion chains. To account for frequency, I also calculate *weighted compliance* as $\sum_{\forall i \in R_p^\star} f(i) / \sum_{\forall j \in R_p} f(j)$, where $f(t)$ is the number of times tuple t appears in RTB ad inclusion chains on p.

5.5.3 Results

Figure 5.12 show the percentage of non-compliant tuples per publisher in our April 2019 snapshot. We notice that for both unweighted and weighted, very few publishers experience high-compliance with respect to their impression inventory. Only 2% of the publishers have 0% non-compliance. Conversely, ≥90% of publishers experience non-compliance for the majority of their (s, b) tuples. Additionally, ≥50% of publishers have 100% non-compliant tuples. Given that ads.txt was introduced in 2017 [167] and is being pushed forcefully by major exchanges like Google [89], I was surprised by the lack of compliance.

Table 5.4: Top 20 non-compliant Seller-Buyer pairs, sorted by presence on number of unique publishers.

Seller	Buyer	# Publishers (%)		Total Chains (%)	
gumgum	domdex	247	20.38	280	16.25
gumgum	appnexus	225	20.49	237	20.10
taboola	weborama	188	52.66	190	51.77
taboola	rubiconproject	154	11.55	404	11.31
dailymotion	dyntrk	148	51.21	1296	42.99
taboola	indexexchange	144	11.61	190	11.59
gumgum	pubmatic	139	27.25	480	28.27
justpremium	openx	138	100.00	936	100.00
criteo	media	120	74.53	454	77.47
rubiconproject	yahoo	120	2.63	120	2.63
criteo	yieldlab	105	78.36	756	80.51
taboola	pubmatic	104	12.87	512	12.41
springserve	pubmatic	103	49.28	4668	53.84
exponential	google	101	31.46	1700	20.83
criteo	ligadx	98	77.78	502	83.11
criteo	pubmatic	84	82.35	415	78.60
nativeroll	weborama	81	100.00	647	100.00
nativeroll	seedr	78	100.00	464	100.00
aniview	google	76	84.44	5047	82.21
yandex	google	65	98.48	1744	97.76

To understand the high non-compliance rate, I manually looked at the most frequent non-compliant tuples. One of the most non-compliant sellers was doubleclick.net, which illustrates a deficiency in my analysis thus far. Recall that Google specifies that google.com is the correct seller domain to place in ads.txt files. This causes us to incorrectly mark sellers like DoubleClick as non-compliant since the domain is not explicitly listed in ads.txt files.

Clustering Domains. To tackle this issue, I clustered domains together that belong to the same organization using data provided by *WhoTracksMe* [192]. This dataset is gathered by *Cliqz*, which is a German company that develops privacy-preserving web browser and extensions [40].[7] This dataset contains mappings for 28 parent domains, including Google, OpenX, Rubicon Project, etc.

Using this dataset, I map the domains that appear in our RTB ad inclusion chains to their parent domain and re-plot the non-compliance in Figure 5.13. This changed the results dramatically. The percentage of publishers with complete compliance rises from 2% in the non-clustered case to 70% in the clustered case. Furthermore, only 3% of publishers experience 100% non-compliance instead of 50%. Compliance rises even further when I filter out (s, b) tuples where $s = b$, i.e., the same domain appears adjacent to itself in the ad inclusion chain. Overall, I can conclude that the vast majority of RTB ads in our dataset appear to have been served by buyers who were in compliance with publishers' ads.txt files.

[7] I provide the list of clustered domains along with their parent domains on https://www.ccs.neu.edu/home/ahmad/thesis/files/clustered_for_adstxt.json.

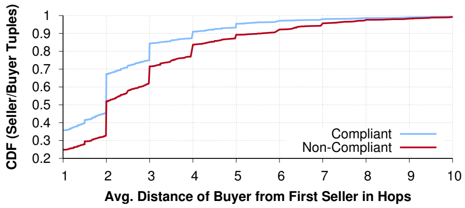


Figure 5.14: Average distance of buyer from first seller across all publishers. Distances are shown for both compliant and non-compliant tuples.

Table 5.5: Percentage of `ads.txt`-enabled publishers on top sellers.

Seller	% Publishers w/ `ads.txt`	# Publishers w/ RTB Ads
google	58.64	23552
advertising	75.46	7196
pubmatic	79.53	6800
rubiconproject	88.37	5562
openx	91.18	3173
appnexus	91.71	3150
sovrn	90.61	2279
indexexchange	88.98	1915
teads	93.99	1232
smartadserver	92.17	1085

Distance. One interesting question is *when do non-compliant ad auctions occur in the inclusion chains?*, i.e., in the seller that directly receives the impression from the publisher, or farther down the chain? Figure 5.14 shows the average distance of the buyer from the very first authorized seller for complaint and non-compliant tuples. We observe a clear separation between the lines, with non-compliant buyers tending to be one hop farther away from the first seller than complaint buyers on average. This confirms our intuition that compliance with the `ads.txt` standard tends to be stronger earlier in chains when top sellers are typically conducting the auctions. In contrast, as the chain length grows, less reputable buyers and sellers become involved, and compliance wanes.

Non-Compliant Sellers. Next, I take a deeper look into the seller and buyer domains from the non-compliant tuples. Table 5.4 shows the top 20 non-compliant tuples across all publishers, after clustering them by their parent domains. For each tuple, I show the total number and percentage of publishers it was non-compliant on. Table 5.4 also shows the total number and percentage of times the tuple was non-compliant across all publishers.

With respect to the non-compliant sellers, several companies appear to be systematically non-compliant, such as NativeRoll, GumGum, Criteo, and JustPremium. Only one of the top authorized sellers from Table 5.3 (Rubicon Project) appears on the list. However, it is only non-compliant with a single buyer and only in 2.6% of transactions in our dataset. This finding suggests that top authorized sellers like Google and OpenX are enforcing compliance with the `ads.txt` standard within their markets.

One possibility is that top sellers are only auctioning impression inventory that can be validated, i.e., from publishers with `ads.txt` files. However, this is not the case: Table 5.5 shows (1) the number of publishers in our dataset that had RTB ad inclusion chains with the given seller, and (2) the percentage of these publishers that had `ads.txt` files. For example, only 59% of the publishers in our dataset whose impression inventory moved through Google's exchange had an `ads.txt` file. This demonstrates that all of the top sellers are, to some extent, still auctioning inventory that cannot be validated using `ads.txt`.

A second possibility is that top sellers are faithfully following the `ads.txt` standard by refusing to auction unauthorized impressions. Although our data suggest that this might be the case, I cannot guarantee this from observational data alone. I attempted to become a publisher to conduct controlled experiments to test compliance with the `ads.txt` standard, but I was unable to do so.[8]

Non-Compliant Buyers. With respect to non-compliant buyers, the striking feature of Table 5.4 is that most are actually SSPs/ad exchanges, including eight of the top authorized sellers from Table 5.3. In other words, top DSPs seem to be following the `ads.txt` standard by not buying non-compliant inventory. Rather, *sellers* are buying non-compliant inventory, although the reason for this is unclear since it seems unlikely that they are able to resell this non-compliant inventory at auction. Many of these companies offer seller- and buyer-side products, so it is possible that they are purchasing this non-compliant inventory and then serving ads, rather than reselling. Still, this behavior is surprising given that many of these companies have called for strict enforcement of the `ads.txt` standard [16, 81, 148, 169].

5.6 Summary

In this chapter, I presented the first large-scale, longitudinal study of the `ads.txt` standard. Using data crawled from 240K websites over a period of 15-months, I examined the adoption of `ads.txt` by publishers, the contents of these files, the characteristics of sellers (ad exchanges) who appear in the files, and compliance with the standard by sellers and buyers.

Transparency. One of the motivating questions of this study was *how useful is* `ads.txt` *as a transparency mechanism?* The answer to this question is mixed. On the positive side, `ads.txt` is enjoying wide adoption. For the first time ever, publishers are explicitly declaring who they have advertising contracts with. Further, by aggregating across `ads.txt` files, it is possible to compile an explicit and extensive list of seller-side advertising platforms. Additionally, coupled with inclusion chain data, buyer-side platforms can also be identified. These datasets are extremely useful for measurement studies of the online ad ecosystem, which historically have had to rely on heuristics or crowdsourced data (e.g., EasyList) to identify these domains. Additionally, this data may be useful

[8] All of the ad exchanges I contacted refused to engage with me unless my website received on the order of millions of unique visitors per month.

for browser extensions that inform users about the advertising practices of publishers [145] or block ads. The list of ad exchanges from this dataset completes an important piece of this dissertation, and I use it in chapter 6 to model how user impressions are propagated in the ad ecosystem via these exchanges.

However, there are several caveats to the `ads.txt` data. First, as we saw throughout this study, `ads.txt` files contain various classes of errors that must be mitigated by consumers of the data. Fortunately, I develop techniques in this study that can help in this regard. Second, `ads.txt` is only designed to make advertising domains transparent, not tracking domains. Additional datasets and detection techniques are still necessary to identify trackers. Finally, I note that the seller domains listed in `ads.txt` files are not all-inclusive; additional, manual work is required to map seller domains like `google.com` to all of the other domains used by sellers.

Compliance. The other motivating questions behind this study was *are members of the online ad ecosystem complying with the* `ads.txt` *standard?* Here again, the answer is somewhat mixed. With respect to adoption, I found that over 60% of popular publishers that are monetized via RTB ads have adopted `ads.txt`, which is impressive for a standard that is just over two years old (as of this writing). Further, our analysis of ad inclusion chains strongly suggests that SSPs and ad exchanges are honoring the standard by not attempting to sell unauthorized inventory. Future work should attempt to validate this using causal experiments.

That said, there is a great deal of room for improvement before domain spoofing will be eradicated. There are still many publishers that have not adopted `ads.txt`, and their impression inventory continues to be purchased from SSPs/ad exchanges. All of these domains are vulnerable to spoofing. Additionally, I do observe specific sellers that continue to sell impressions that they are not authorized to sell, as well as specific buyers (including many top ad exchanges) who continue to purchase impressions from these unauthorized sellers. All of these companies run the risk of introducing spoofed inventory into the marketplace.

Chapter 6

Diffusion of User Tracking Data in the Online Advertising Ecosystem

The rise of Real Time Bidding (RTB) has forced Advertising and Analytics (*A&A*) companies to collaborate more closely with one another, in order to exchange data about users and facilitate bidding on impressions [21, 147]. In § 1.1, I explained that user information can be shared under RTB via two primary means: (1) A&A domains actively share user identifiers with each other through *cookie matching* to facilitate RTB, and (2) ad exchanges disperse tracking information to multiple DSPs during RTB auctions to solicit bids for user impressions. The primary goal of my dissertation is to study the privacy implications of RTB so that we can better understand the privacy digital footprint of the user in the modern ad ecosystem.

In chapter 4, I proposed a generic methodology to detect information-sharing among A&A domains which facilitate RTB auctions. And, in chapter 5, I conduct a measurement study of the `ads.txt` transparency standard to collect a list of ad exchanges involved in selling impressions during RTB auctions. Now, we are ready to use these results to come up with a model that captures the effect of RTB on users' privacy. However, due to the enormous complexity of the ad ecosystem and close collaboration among A&A domains, we cannot accurately determine the extent of privacy leakage if we look at RTB auctions in isolation.

A natural way to model this complex ecosystem is in the form of a graph. Graph models that accurately capture the relationships between publishers and A&A companies are extremely important for practical applications, such as estimating revenue of A&A companies [74], predicting whether a given domain is a tracker [102], or evaluating the effectiveness of domain-blocking strategies on preserving users' privacy.

However, to date, technical limitations have prevented researchers from developing accurate graph models of the online advertising ecosystem. For example, Gomer et al. [78] propose a *Referer* graph, where nodes represent publishers or A&A domains, and two nodes a_i and a_j are connected if an HTTP message to a_j is observed with a_i as the HTTP `Referer`. Unfortunately, as I will show, graphs built using `Referer` information may contain erroneous edges in cases where a third-party script is embedded directly into a first-party context (i.e., is not sandboxed in an `iframe`).

In this chapter, I use the data collected from the methodology proposed in chapter 4, and the list of ad exchanges from `ads.txt` standard in chapter 5 to model the diffusion of user tracking data within RTB auctions. In particular, I propose a novel and accurate representation of the advertising graph called an *Inclusion* graph. The *Inclusion* graph corrects the technical problem of the *Referer* graph by using the actual inclusion relationships between domains to represent edges, rather than imprecise `Referer` relationships. I am able to construct *Inclusion* graphs, thanks to the advances in browser instrumentation that allow researchers to conduct web crawls, including the ones in chapter 4 and chapter 5, that record the exact provenance of all HTTP(S) requests [17, 21, 112].

I use crawled data from chapter 4, consisting of around 2M impressions from popular e-commerce websites to construct the *Inclusion* graph. In § 6.3, I examine the fundamental graph properties of the *Inclusion* graph and compare it to a *Referer* graph, created using the same dataset to understand their salient differences. In § 6.4, I demonstrate a concrete use case for the *Inclusion* graph by using simulations to model the flow of tracking data to A&A companies. Furthermore, I compare the efficacy of different real-world and graph-theoretic "blocking" strategies (e.g., AdBlock Plus [7], Ghostery [73], and Disconnect [52]) at reducing the flow of tracking information to A&A companies.

Overall, in this chapter, I make the following key contributions:

- I introduce the *Inclusion* graph as a model for capturing the complexity of the online advertising ecosystem. I use the *Inclusion* graph as a substrate for modeling the flow of impressions to A&A companies by taking into account the browsing behavior of users and the dynamics of RTB auctions.
- I find that the *Inclusion* graph has substantive differences in graph structure compared to the *Referer* graph because 48.4% of resource inclusions in our crawled data have an inaccurate `Referer`.
- Through simulations, I find that top 10% of A&A domains are each able to observe more than

90% of an average user's impressions as they browse, under modest assumptions about data sharing in RTB auctions. This includes expected companies like Google and unexpected ones like Pinterest. 636 A&A domains can observe at least 50% of an average user's impressions. Even under the strictest simulation assumptions, the top 10 A&A domains observe 89-99% of all user impressions.

- I simulate the effect of five blocking strategies and find that AdBlock Plus (the world's most popular ad-blocking browser extension [122, 159], is ineffective at protecting users' privacy because major ad exchanges are whitelisted under the Acceptable Ads program [190]. In contrast, Disconnect blocks the most information flows to A&A companies, followed by the removal of top 10% A&A nodes. However, even with strong blocking, major A&A companies still observe 40–80% of user impressions.

6.1 Related Work on Graph Models

In this section, I will briefly go over prior work that modeled the advertising ecosystem as a graph. There are mainly two different models that have been proposed; a *Referer* graph by Gomer et al. [78] and a *Co-occurrence* graph by Kalavri et al. [102].

Gomer et al. [78] built and analyzed graphs of the ad ecosystem by making use of the *Referer* field from HTTP requests. In this representation, a relationship $d_i \rightarrow d_j$ exists if there is an HTTP request to domain d_j with a *Referer* header from domain d_i. While the *Referer* graph provided interesting insights into the structure of the ad ecosystem, its referral-based representation has a significant limitation. As I describe in § 6.2.2, relying on the HTTP *Referer* **does not** always capture the correct relationships between A&A parties, thus leading to incorrect graphs of the ad ecosystem. I re-create this graph representation using our dataset (see § 6.2.2) and compare its properties to a more accurate representation I propose in § 6.3.

Kalavri et al. [102] created a bipartite graph of publishers and associated A&A domains, then transformed it to create an undirected graph consisting solely of A&A domains. In their representation, two A&A domains are connected if they were included by the same publisher. This construction leads to a highly dense graph with many complete cliques. Kalavri et al. leveraged the tight community structure of A&A domains to predict whether new, unknown URLs belong to A&A domains or not. However, this co-occurrence representation has a conceptual shortcoming: it may include edges between A&A domains that do not directly communicate or have any business relationship with. Due to this shortcoming, I do **not** explore this graph representation in my work.

6.2 Methodology

My goal is to capture the most accurate representation of the online advertising ecosystem, which will then allow me to model the effect of RTB on the diffusion of user tracking data. In this section, I introduce the dataset used in this study and describe how I use it to build a graph representation of the ad ecosystem.

6.2.1 Dataset

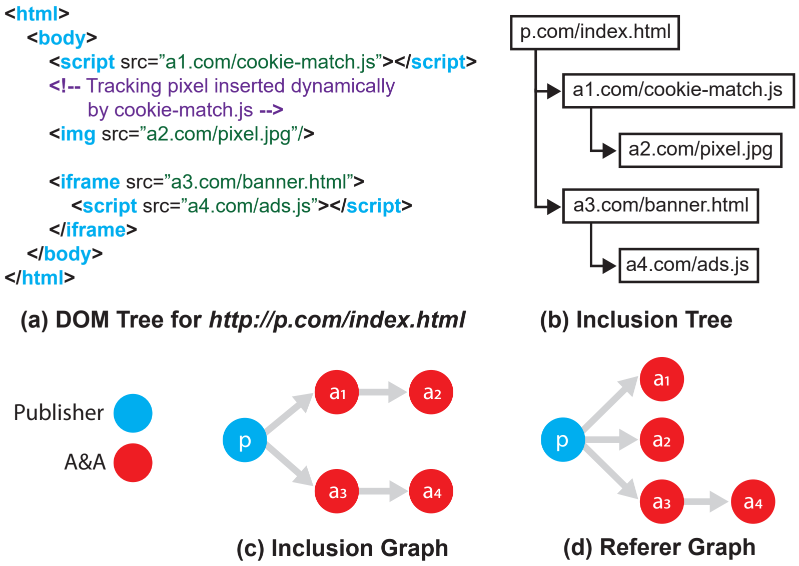
In this study, I use the dataset from chapter 4, where the goal was to causally infer the information-sharing relationships between A&A domains by (1) crawling products from popular e-commerce websites and then (2) observing corresponding *retargeted* ads on publishers. There, I conducted web crawls that covered 738 major e-commerce websites (e.g., Amazon)[1], and 150 popular publishers (e.g., CNN)[2] manually chosen from the Alexa Top-1K. I first crawled 10 manually selected products per e-commerce site to signal strong *intent* to trackers and advertisers, and then crawl 15 randomly chosen pages per publisher to elicit display ads. In total, I repeated the entire crawl nine times and collected around 2M impressions.

Inclusion Trees. This data was collected using a specially instrumented version of Chromium (see § 4.1.2). This tool allows us to record the *inclusion tree* for each webpage, which is a data structure that captures the semantic relationships between elements in a webpage (as opposed to the DOM, which captures syntactic relationships) [17, 112]. The crawler also recorded all HTTP request and response headers associated with each visited URL.

To illustrate the importance of inclusion trees, consider the example webpage shown in Figure 6.1(a). The DOM shows that the page from publisher p ultimately includes resources from four third-party domains (a_1 through a_4). It is clear from the DOM that the request to a_3 is responsible for causing the request to a_4, since the `script` inclusion is within the `iframe`. However, it is not clear which domain generated the requests to a_2 and a_3: the `img` and `iframe` could have been embedded in the original HTML from p, or these elements could have been created dynamically by the `script` from a_1. Although the HTML comment gives us a clue about the provenance of a_2, this information is not captured in the DOM, nor is it obvious how to programmatically extract this information. In this case, the inclusion tree shown in Figure 6.1(b) reveals that the `image` from a_2 was dynamically created by the `script` from a_1, while the `iframe` from a_3 was embedded

[1] http://www.alexa.com/topsites/category/Top/Shopping

[2] For simplicity, I refer to these e-commerce websites as publishers, to distinguish them from A&A domains.



```
<html>
  <body>
    <script src="a1.com/cookie-match.js"></script>
    <!-- Tracking pixel inserted dynamically
         by cookie-match.js -->
    <img src="a2.com/pixel.jpg"/>

    <iframe src="a3.com/banner.html">
      <script src="a4.com/ads.js"></script>
    </iframe>
  </body>
</html>
```

(a) DOM Tree for *http://p.com/index.html*

(b) Inclusion Tree

(c) Inclusion Graph

(d) Referer Graph

Figure 6.1: An example HTML document and the corresponding inclusion tree, *Inclusion* graph, and *Referer* graph. In the DOM representation, the a_1 `script` and a_2 `img` appear at the same level of the tree; in the inclusion tree, the a_2 `img` is a child of the a_1 `script` because the latter element created the former. The *Inclusion* graph has a 1:1 correspondence with the inclusion tree. The *Referer* graph fails to capture the relationship between the a_1 `script` and a_2 `img` because they are both embedded in the first-party context, while it correctly attributes the a_4 `script` to the a_3 `iframe` because of the context switch.

directly in the HTML from p. Note that if `Referer` headers had been used instead, the request to a_2 would have been misattributed to p, since a_1's JavaScript is included in the first-party context.

Cookie Matching. The dataset also includes labels on edges of the inclusion trees, indicating cases where cookie matching is occurring. These labels are derived from heuristics (e.g., string matching to identify the passing of cookie values in HTTP parameters) and causal inferences based on the presence of retargeted ads. In total, our dataset includes 200 empirically validated pairs of A&A domains that match cookies. I use these pairs in § 6.4 to constrain some of the simulations in my models.

6.2.2 Graph Construction

A natural way to model the online ad ecosystem is by using a graph. In this model, nodes represent A&A domains, publishers, or other online services. Edges capture relationships between these actors, such as resource inclusion or information flow (e.g., cookie matching).

Canonicalizing Domains. I use the data described in § 6.2.1 to construct a graph for the online advertising ecosystem. I use effective 2^{nd}-level domain names to represent nodes. For example, `x.doubleclick.net` and `y.doubleclick.net` are represented by a single node labeled `doubleclick`. Throughout this study, when I say "domain", I am referring to an effective 2^{nd}-level domain name.[3]

Simplifying domains to the effective 2^{nd}-level is a natural encoding for advertising data. Consider two inclusion trees generated by visiting two publishers: publisher p_1 forwards the impression to `x.doubleclick.net` and then to advertiser a_1. Publisher p_2 forwards to `y.doubleclick.net` and advertiser a_2. This does not imply that `x.doubleclick` and `y.doubleclick` only sell impressions to a_1 and a_2, respectively. In reality, DoubleClick is a single auction, regardless of the subdomain, and a_1 and a_2 have the opportunity to bid on all impressions. Individual inclusion trees are snapshots of how one particular impression was served; **only in aggregate can all participants in the auctions be enumerated**. Further, 3^{rd}-level domains may read 2^{nd}-level cookies without violating the Same Origin Policy [135]: `x.doubleclick.com` and `y.doubleclick.com` may both access cookies set by `.doubleclick`, and do so in practice.

The sole exception to the domain canonicalization process is Amazon's Cloudfront Content Delivery Network (CDN). I routinely observed Cloudfront hosting ad-related scripts and images in the data. I manually examined the 50 fully-qualified Cloudfront domains that were pre- or proceeded by A&A domains in the data and mapped each one to the corresponding A&A domain. For example, `d31550gg7drwar.cloudfront.net` gets mapped to `adroll`.

***Inclusion* graph.** I propose a novel representation called an *Inclusion* graph that is the union of all inclusion trees in our dataset. This representation is a directed graph of publishers and A&A domains. An edge $d_i \rightarrow d_j$ exists if we have ever observed domain d_i including a resource from d_j. Edges may exist from publishers to A&A domains, or between A&A domains. Figure 6.1(c) shows an example *Inclusion* graph.

***Referer* graph.** Gomer et al. [78] also proposed a directed graph representation consisting of publishers and A&A domains for the online advertising ecosystem. In this representation, each publisher and A&A domain is a node, and edge $d_i \rightarrow d_j$ exists if we have ever observed an HTTP request to d_j with `Referer` d_i. Figure 6.1(d) shows an example *Referer* graph corresponding to the given webpage. Our dataset also includes all HTTP request and response headers from the crawl, and we use these to construct the *Referer* graph.

[3] None of the publishers and A&A domains in our dataset have two-part TLDs, like `.co.uk`, which simplifies the analysis.

Although the *Referer* and *Inclusion* graphs seem similar, they are fundamentally different for technical reasons. Consider the examples shown in Figure 6.1: the `script` from a_1 is included directly into p's context, thus p is the `Referer` in the request to a_2. This results in a *Referer* graph with two edges that does **not** correctly encode the relationships between the three parties: $p \rightarrow a_1$ and $p \rightarrow a_2$. In other words, HTTP `Referer` headers are an indirect method for measuring the semantic relationships between page elements, and the headers may be incorrect depending on the syntactic structure of a page. The *Inclusion* graph representation fixes the ambiguity in the *Referer* graph by explicitly relying on the inclusion relationships between elements in webpages. I analyze the salient differences between the *Referer* and *Inclusion* graph in § 6.3.

Weights. Additionally, I also create a weighted version of these graphs. In the *Inclusion* graph, the weight of $d_i \rightarrow d_j$ encodes the number of times a resource from d_i sent an HTTP request to d_j. In the *Referer* graph, the weight of $d_i \rightarrow d_j$ encodes the number of HTTP requests with `Referer` d_i and destination d_j.

6.2.3 Detection of A&A Domains

For us to understand the role of A&A companies in the advertising graph, we must be able to distinguish A&A domains from publishers and non-A&A third parties like CDNs. In the *inclusion trees* in our dataset, each resource is labeled as A&A or non-A&A using the EasyList [54] and EasyPrivacy [55] rule lists. For all the A&A labeled resources, I extract the associated 2^{nd}-level domains. To eliminate false positives, I only consider a 2^{nd}-level domain to be A&A if it was labeled as A&A more than 10% of the time in the dataset.

6.2.4 Coverage

There are two potential concerns with the raw data I use in this study: *does the data include a representative set of A&A domains?* and *does the data contain all of the outgoing edges associated with each A&A domain?* To answer the former question, I plot Figure 4.2 (see § 4.1), which shows the overlap between the top x A&A domains in our dataset (ranked by inclusion frequency by publishers) with all of the A&A domains included by the Alexa Top-5K websites.[4] I observe that 99% of the 150 most frequent A&A domains appear in both samples, while 89% of the 500 most frequent appear in both. These findings confirm that our dataset includes the vast majority of prominent A&A domains that users are likely to encounter on the web.

[4] Our dataset and the Alexa Top-5K data were both collected in December 2015, so they are temporally comparable.

Table 6.1: Basic statistics for *Inclusion* and *Referer* graph. I show sizes for the largest WCC in each graph. † denotes that the metric is calculated on the largest SCC. ‡ denotes that the metric is calculated on the undirected transformation of the graph.

					Avg. Deg.		Avg. Path	Cluster.		Degree
Graph Type	**|V|**	**|E|**	**$\|V_{WCC}\|$**	**$\|E_{WCC}\|$**	**(In**	**Out)**	**Length**	**Coef.**	**S^{Δ} [93]**	**Assort.**
Inclusion	1917	26099	1909	26099	13.612	13.612	2.748†	0.472‡	31.254‡	-0.31‡
Referer	1923	41468	1911	41468	21.564	21.564	2.429†	0.235‡	10.040‡	-0.29‡

To answer the second question, I plot Figure 4.3 (see § 4.1), which shows the number of unique external A&A domains contacted by A&A domains in our dataset as the crawl progressed (i.e., starting from the first page crawled, and ending with the last). Recall that the dataset was collected over nine consecutive crawls spanning two weeks, each of which visited 9,630 individual pages spread over 888 domains.

I observe that the number of A&A →A&A edges rises quickly initially, going from 0 to 800 in 3,600 crawled pages. Then, the growth slows down, requiring an additional 12,000 page visits to increase from 800 to 900. In other words, almost all A&A edges were discovered by half-way through the very first crawl; eight subsequent iterations of the crawl only uncovered 12.5% more edges. This demonstrates that the crawler reached the point of diminishing returns, indicating that the vast majority of connections between A&A domains that existed at the time are contained in the dataset.

6.3 Graph Analysis

In this section, I look at the essential graph properties of the *Inclusion* graph. This sets the stage for a higher-level evaluation of the *Inclusion* graph in § 6.4.

6.3.1 Basic Analysis

I begin by discussing the basic properties of the *Inclusion* graph, as shown in Table 6.1. For reference, I also compare the properties with those of the *Referer* graph.

Edge Misattribution in the *Referer* graph. The *Inclusion* and *Referer* graph have essentially the same number of nodes, however, the *Referer* graph has 159% more edges. I observe that **48.4% of resource inclusions in the raw dataset have an inaccurate `Referer`** (i.e., the first-party is the `Referer` even though the resource was requested by third-party JavaScript), which is the cause for additional edges in the *Referer* graph.

There is a massive shift in the location of edges between the *Inclusion* and the *Referer* graph: the number of publisher → A&A edges decreases from 33,716 in the *Referer* graph to 10,274 in the *Inclusion* graph, while the number of A&A → A&A edges increases from 7,408 to 13,546. In the *Referer* graph only 3% of A&A → A&A edges are reciprocal, versus 31% in the *Inclusion* graph. Taken together, these findings highlight the practical consequences of misattributing edges based on `Referer` information, i.e., relationships between A&A companies that should be in the core of the network are incorrectly attached to publishers along the periphery.

Structure and Connectivity. As shown in Table 6.1, the *Inclusion* graph has large, well-connected components. The largest Weakly Connected Component (WCC) covers all but eight nodes in the *Inclusion* graph, meaning that very few nodes are completely disconnected. This highlights the inter-connectedness of the ad ecosystem. The average node degree in the *Inclusion* graph is 13.6, and $<$7% of the nodes have in- or out-degree $\geq$50. This result is expected: publishers typically only form direct relationships with a small number of SSPs and exchanges, while DSPs and advertisers only need to connect to the major exchanges. The small number of high-degree nodes are ad exchanges, ad networks, trackers (e.g., Google Analytics), and CDNs.

The *Inclusion* graph exhibits a low average shortest path length of 2.7, and a very high average clustering coefficient of 0.48, implying that it is a "small world" graph. I show the "small-worldness" metric $S^{\triangle}$ in Table 6.1, which is computed for a given undirected graph G and an equivalent random graph G_R[5] as $S^{\triangle} = (C^{\triangle}/C^{\triangle}_{R})/(L^{\triangle}/L^{\triangle}_{R})$, where $C^{\triangle}$ is the average clustering[6] coefficient, and $L^{\triangle}$ is the average shortest path length [93]. The *Inclusion* graph has a large $S^{\triangle} \approx 31$, confirming that it is a "small world" graph.

Lastly, Table 6.1 reveals that both *Inclusion* and *Referer* graphs are disassortative (i.e., low degree nodes tend to connect to high degree nodes).

Change Over Time. Our *Referer* graph exhibits interesting differences compared to the *Referer* graph examined by Gomer et al. [78], which I constructed based on crawled data from 2013. Specifically, *Referer* graph constructed by me has higher average node degree (21.728 vs. 8.796), higher average clustering coefficient (0.239 vs. 0.196), and lower average shortest path lengths (2.429 vs. 3.673).[7] This demonstrates that the ad network graph is densifying over time.

Summary. My measurements demonstrate that the structure of the ad network graph is troubling

[5] Equivalence in this case means that for G and G_R, $|V| = |V_R|$ and $|E|/|V| = |E_R|/|V_R|$.

[6] I compute the average clustering by transforming directed graphs into undirected graphs, and compute average shortest path lengths on the SCC.

[7] To ensure a fair comparison, I compare my *Referer* graph to the U.S.-based *Referer* graph from [78].

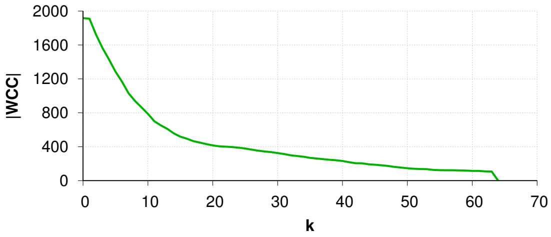


Figure 6.2: k-core: size of the *Inclusion* graph WCC as nodes with degree $\leq k$ are recursively removed.

from a privacy perspective. Short path lengths and high clustering between A&A domains suggest that the data tracked from users will spread rapidly to all participants in the ecosystem (I examine this in more detail in § 6.4). This rapid spread is facilitated by high-degree hubs in the network that have disassortative connectivity, which we examine in the next section. Furthermore, comparisons with historical graphs collected in 2013 suggest that the ad network is getting denser, with more connections between A&A domains. This suggests that the user data can spread more widely and more quickly than in the past, which is also concerning.

6.3.2 Cores and Communities

I now examine how nodes in the *Inclusion* graph connect to each other using two metrics: k-cores and community detection. The k-core of a graph is the subset of a graph (nodes and edges) that remain after recursively removing all nodes with degree $\leq k$. By increasing k, the loosely connected periphery of a graph can be stripped away, leaving just the dense core. In our scenario, this corresponds to the high-degree ad exchanges, ad networks, and trackers that facilitate the connections between publishers and advertisers.

Figure 6.2 plots k versus the size of the WCC for the *Inclusion* graph. The plot shows that the core of the *Inclusion* graph rapidly declines in size as k increases, which highlights the interdependence between A&A domains and the lack of a distinct core.

Next, to examine the community structure of the *Inclusion* graph, I utilized three different community detection algorithms: label propagation by Raghavan et al. [161], Louvain modularity maximization [26], and the centrality-based Girvan-Newman [75] algorithm. I chose these algorithms because they attempt to find communities using fundamentally different approaches.

Unfortunately, after running these algorithms on the largest WCC, the results of my community-detection analysis were negative. Label propagation clustered all nodes into a single community.

Table 6.2: Top 10 nodes ranked by betweenness centrality and weighted PageRank in the *Inclusion* graph.

Betweenness Centrality	**Weighted PageRank**
google-analytics	doubleclick
doubleclick	googlesyndication
googleadservices	2mdn
facebook	adnxs
googletagmanager	google
googlesyndication	adsafeprotected
adnxs	google-analytics
google	scorecardresearch
addthis	krxd
criteo	rubiconproject

Louvain found 14 communities with an overall modularity score of 0.44 (on a scale of -1 to 1 where 1 is entirely disjoint clusters). The largest community contains 771 nodes (40% of all nodes) and 3252 edges (12% of all edges). Out of 771 nodes, 37% are A&A. However, none of the 14 communities corresponded to meaningful groups of nodes, either segmented by type (e.g., publishers, SSPs, DSPs, etc.) or segmented by ad exchange (e.g., customers and partners centered around DoubleClick). This is a known deficiency in modularity maximization based methods, that they tend to produce communities with no real-world correspondence [13]. Girvan-Newman found 10 communities, with the largest community containing 1,097 nodes (57% of all nodes) and 16,424 edges (63% of all edges). Out of 1,097 nodes, 64% are A&A. However, the modularity score was zero, which means that the Girvan-Newman communities contain a random assortment of internal and external (cross-cluster) edges.

These results show exactly how challenging it is to determine the role of A&A domains, and that is the reason why I utilized `ads.txt` standard in chapter 5 to isolate a list for A&A domains which act as ad exchanges. Overall, these results demonstrate that the web display ad ecosystem is not balkanized into distinct groups of companies and publishers that partner with each other. Instead, the ecosystem is highly interdependent, with no clear delineations between groups or types of A&A companies. This result is not surprising considering how dense the *Inclusion* graph is.

6.3.3 Node Importance

In this section, I focus on the importance of specific nodes in the *Inclusion* graph using two metrics: betweenness centrality and weighted PageRank. As before, I focus on the largest WCC. The betweenness centrality for a node n is defined as the fraction of all shortest paths on the graph that traverse n. In our scenario, nodes with high betweenness centrality represent the key pathways for tracking information and impressions to flow from publishers to the rest of the ad ecosystem. For weighted PageRank, I weight each edge in the *Inclusion* graph based on the number of times we observe it in our raw data. In essence, weighted PageRank identifies the nodes that receive the largest amounts of tracking data and impressions throughout each graph.

Table 6.2 shows the top 10 nodes in the *Inclusion* graph based on betweenness centrality and weighted PageRank. Prominent online advertising companies are well represented, including AppNexus (*adnxs*), Facebook, and Integral Ad Science (*adsafeprotected*). Similar to prior work, I find that Google's advertising domains (including DoubleClick and *2mdn*) are the most prominent overall [78]. Unsurprisingly, these companies all provide platforms, i.e., SSPs, ad exchanges, and ad networks. We also observe trackers like Google Analytics and Tag Manager. Interestingly, among 14 unique domains across the two lists, ten only appear in a single list. This suggests that the most important domains in terms of connectivity are not necessarily the ones that receive the highest volume of HTTP requests.

6.4 Information Diffusion

In § 6.3, I examined the descriptive characteristics of the *Inclusion* graph and discussed the implications of this graph structure on our understanding of the online advertising ecosystem. In this section, I take the next step and present a concrete use case for the *Inclusion* graph: modeling the diffusion of user tracking data across the ad ecosystem under different types of ad and tracker blocking (e.g., AdBlock Plus and Ghostery). I model the flow of information across the *Inclusion* graph, taking into account different blocking strategies, as well as the design of RTB systems and empirically observed transition probabilities from our crawled dataset.

6.4.1 Simulation Goals

Simulations are an important tool for helping to understand the dynamics of the (otherwise opaque) online advertising industry. For example, Gill et al. used data-driven simulations to model the distribution of revenue amongst online display advertisers [74].

Here, I use simulations to examine the flow of browsing history data to trackers and advertisers. Specifically, I ask:

1. How many user impressions (i.e., page visits) on publishers can each A&A domain observe?

2. What fraction of the unique publishers that a user visits can each A&A domain observe?
3. How do different blocking strategies impact the number of impressions and fraction of publishers observed by each A&A domain?

These questions have direct implications for understanding users' online privacy. The first two questions are about quantifying a user's online digital footprint, i.e., how much of their browsing history can be recorded by different companies. In contrast, the third question investigates how well different blocking strategies perform at protecting users' privacy.

6.4.2 Simulation Setup

To answer these questions, I simulate the browsing behavior of users using the model provided by Burklen et al. [31].[8] In particular, I simulate a user browsing publishers over discreet time steps. At each time step our simulated user decides whether to remain on the current publisher according to a Pareto distribution (exponent $= 2$), in which case they generate a new impression on that publisher. Otherwise, the user browses to a new publisher, which is chosen based on a Zipf distribution over the Alexa ranks of the publishers. Burklen et al. developed this browsing model based on large-scale observational traces, and derive the distributions and their parameters empirically. This browsing model has been successfully used to drive simulated experiments in other work [111].

I generated browsing traces for 200 users. On average, each user generated 5,343 impressions on 190 unique publishers. The publishers are selected from the 888 unique first-party websites in our dataset (see § 6.2.1).

During each simulated time step the user generates an impression on a publisher, which is then forwarded to all A&A domains that are directly connected to the publisher. This emulates a webpage with multiple slots for display ads, each of which is serviced by a different SSP or ad exchange. However, it is insufficient to simply forward the impression to the A&A domains directly connected to each publisher; we must also account for ad exchanges and RTB auctions [21, 147], which may cause the impression to spread farther on the graph. I discuss this process next. The simulated time step ends when all impressions arrive at A&A domains that do not forward them. Once all outstanding impressions have terminated, time increments and our simulated user generates a new impression, either from their currently selected publisher or from a new publisher.

[8] To the best of my knowledge, there are no other empirically validated browsing models besides [31].

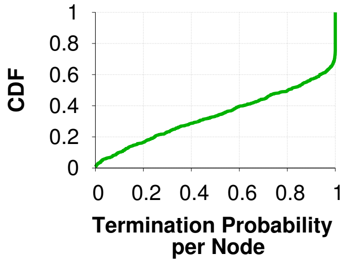


Figure 6.3: CDF of the *termination probability* for A&A nodes.

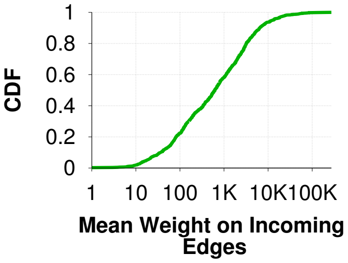


Figure 6.4: CDF of the mean weight on incoming edges for A&A nodes.

6.4.2.1 Impression Propagation

Our simulations must account for *direct* and *indirect* propagation of impressions. Direct flows occur when one A&A domain sells or redirects an impression to another A&A domain. I refer to these flows as "direct" because they are observable by the web browser, and are thus recorded in our dataset. Indirect flows occur when an ad exchange solicits bids on an impression. The advertisers in the auction learn about the impression, but this is not directly observable to the browser; only the winner is ultimately known.

Direct Propagation. To account for direct propagation, I assign a *termination probability* to each A&A node in the *Inclusion* graph that determines how often it serves an ad itself, versus selling the impression to a partner (and redirecting the user's browser accordingly). I derive the termination probability for each A&A node empirically from our dataset. When an impression is sold, I determine which neighboring node purchases the impression based on the weights of the outgoing edges. For a node a_i, I define its set of outgoing neighbors as $\mathcal{N}_o(a_i)$. The probability of selling to neighbor $a_j \in \mathcal{N}_o(a_i)$ is $w(a_i \to a_j)/\sum_{\forall a_y \in \mathcal{N}_o(a_i)} w(a_i \to a_y)$, where $w(a_i \to a_j)$ is the weight of the given edge.

Figure 6.3 shows the *termination probability* for A&A nodes in the *Inclusion* graph. We see that 25% of the A&A nodes have a termination probability of one, meaning that they never sell impressions. The remaining 75% of A&A nodes exhibit a wide range of termination probabilities, corresponding to different business models and roles in the ad ecosystem. For example, DoubleClick, the most prominent ad exchange, has a termination probability of 0.35, whereas Criteo, a well-known advertiser specializing in retargeting, has a termination probability of 0.63.

Figure 6.4 shows the mean incoming edge weights for A&A nodes in the *Inclusion* graph. We observe that the distribution is highly skewed towards nodes with extremely high average incoming weights (note that the x-axis is in log scale). This demonstrates that heavy-hitters like DoubleClick, GoogleSyndication, OpenX, and Facebook are likely to purchase impressions that go up for auction in our simulations.

Indirect Propagation. At the time of this study, there was no way to systematically determine which A&A domains are ad exchanges, or which pairs of A&A domains share information. Because of that, precise accounting for indirect propagation was not possible. To compensate, I evaluated three different indirect impression propagation models. Later, through my analysis of the `ads.txt` standard in chapter 5, I isolate the list of A&A domains $E_{\texttt{ads.txt}}$ that act as ad exchanges and incorporate that into my model.

Following are the three different models I evaluate for indirect propagation:

- **Cookie Matching-Only:** As I highlight in § 6.2.1, our dataset includes 200 empirically validated pairs of A&A domains that match cookies. In this model, I treat these 200 edges as ground-truth and only indirectly disseminate impressions along these edges. Specifically, if a_i observes an impression, it will indirectly share with a_j iff $a_i \rightarrow a_j$ exists and is in the set of 200 known cookie matching edges. This is the **most conservative model** I evaluate, and it provides a lower-bound on impressions observed by A&A domains.
- **RTB Relaxed:** In this model, I assume that each A&A domain that observes an impression, indirectly shares it with all A&A domains that it is connected to. Although this is the correct behavior for ad exchanges like Rubicon and DoubleClick, it is not correct for every A&A domain. This is the **most liberal model** I evaluate, and it provides an upper-bound on impressions observed by A&A domains.
- **RTB Constrained:** In this model, I select a subset of A&A domains E to act as ad exchanges. Whenever an A&A domain in E observes an impression, it shares it with all directly connected A&A domains, i.e., to solicit bids. This model represents a **more realistic** view of information diffusion than the Cookie Matching-Only and RTB Relaxed models because the graph contains few but extremely well-connected exchanges.

Figure 6.5 shows hypothetical examples of how impressions disseminate under my indirect models. Figure 6.5(a) presents the scenario: a graph with two publishers connected to two ad exchanges and five advertisers. a_2 is a bidder in both exchanges and serves as a DSP for a_4 and a_5 (i.e., it services their ad campaigns by bidding on their behalf). Light grey edges capture cases where the two endpoints have been observed cookie matching in the ground-truth data. Edge $e_2 \rightarrow a_3$ is a false

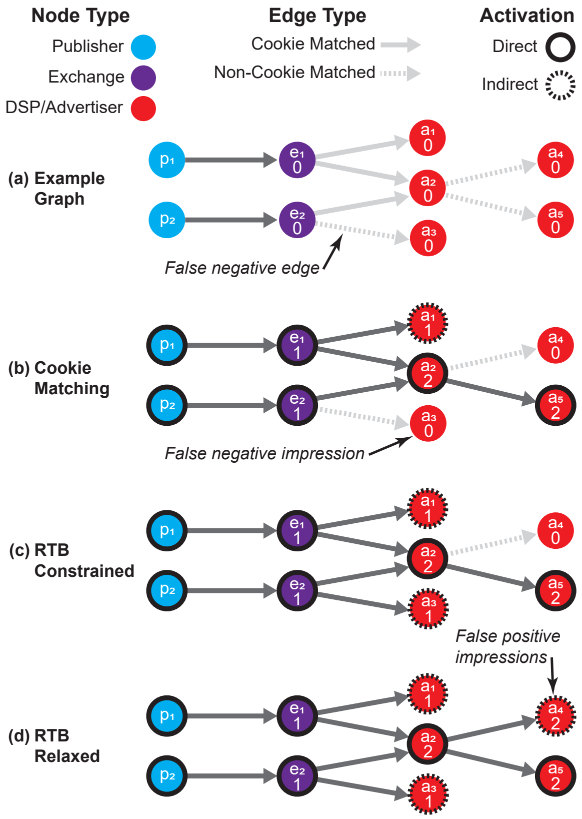


Figure 6.5: Examples of my information diffusion simulations. The observed impression count for each A&A node is shown below its name. **(a)** shows an example graph with two publishers and two ad exchanges. Advertisers a_1 and a_3 participate in the RTB auctions, as well as DSP a_2 that bids on behalf of a_4 and a_5. **(b)–(d)** show the flow of data (dark grey arrows) when a user generates impressions on p_1 and p_2 under three diffusion models. In all three examples, a_2 purchases both impressions on behalf of a_5, thus they both *directly* receive information. Other advertisers *indirectly* receive information by participating in the auctions.

negative because matching has not been observed along this edge in the data, but a_3 must match with e_2 to meaningfully participate in the auction.

Figure 6.5(b)–(d) show the flow of impressions under our three models. In all three examples, a user visits publishers p_1 and p_2, generating two impressions. Further, in all three examples a_2 wins both auctions on behalf of a_5; thus e_1, e_2, a_2, and a_5 are guaranteed to observe impressions. As shown in the figure, a_2 and a_5 observe both impressions, but other nodes may observe zero or more impressions depending on their position and the dissemination model. In Figure 6.5(b), a_3 does not observe any impressions because its incoming edge has not been labeled as cookie matched; this is a false negative because a_3 participates in e_2's auction. Conversely, in Figure 6.5(d), all nodes

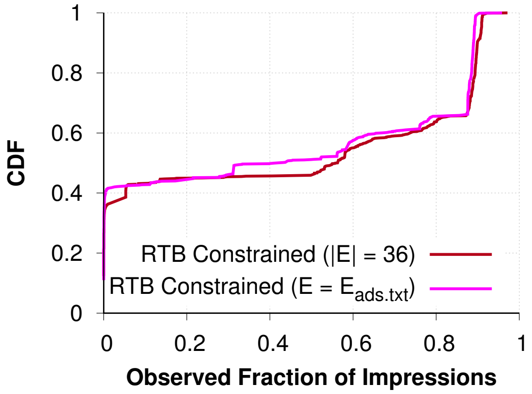


Figure 6.6: Fraction of impressions observed by A&A domains for RTB Constrained model under $|E| = 36$ and $E = E_{\texttt{ads.txt}}$.

always share all impressions, thus a_4 observes both impressions. However, these are false positives, since DSPs like a_2 do not routinely share information amongst all their clients.

Selecting E for RTB Constrained. As I highlighted earlier, at the time of this study, I did not have access to $E_{\texttt{ads.txt}}$ (list of ad exchanges from the `ads.txt` study). So, at the time of the study, I used heuristics to select a subset of A&A domains as E for RTB Constrained. In particular, I selected all A&A nodes with out-degree ≥ 50 and in/out-degree ratio r in the range $0.7 \leq r \leq 1.7$ to be in E. These thresholds were chosen after manually looking at the degrees and ratios for known ad exchanges (e.g., DoubleClick, OpenX, *etc.*) in chapter 4. This resulted in $|E| = 36$ A&A nodes being chosen as ad exchanges (out of 1,032 total A&A domains in the *Inclusion* graph). I enforced restrictions on r because A&A nodes with disproportionately large amounts of incoming edges are likely to be trackers (information enters but is not forwarded out), while those with disproportionately large amounts of outgoing edges are likely SSPs (they have too few incoming edges to be an ad exchange). Table 8.1 in the appendix shows the domains in E, including major, known ad exchanges like App Nexus, Advertising.com, Casale Media, DoubleClick, Google Syndication, OpenX, Rubicon, Turn, and Yahoo. 150 of the 200 known cookie matching edges in our dataset are covered by this list of 36 nodes[9].

It is quite possible that E contains false positives (A&A domains that are not ad exchanges but are added in E) and false negatives (ad exchanges not added in E). This can introduce potential errors in our RTB Constrained simulations. I improve upon this by using $E_{\texttt{ads.txt}}$ as E (i.e., $E = E_{\texttt{ads.txt}}$), where $E_{\texttt{ads.txt}}$ is a list of A&A domains that act as ad exchanges in the `ads.txt`

[9] In the original publication of this study, all the analysis about RTB Constrained was done using $|E| = 36$ [25].

dataset (see chapter 5). Figure 6.6 shows the fraction of total impressions (out of ∼5,300) observed by A&A domains for RTB Constrained when 36 domains are manually selected as E ($|E| = 36$) using the heuristics described above, and when $E = E_{\texttt{ads.txt}}$. We can observe that the two distributions are quite similar, i.e., A&A domains observe a similar fraction of impressions under the two sets of ad exchanges. This is surprising since $E_{\texttt{ads.txt}}$ contains 1035 ad exchanges; far more than $|E| = 36$. One would expect A&A domains to observe more impressions under the $E_{\texttt{ads.txt}}$ distribution since it is more permissive. If anything, A&A domains observe slightly fewer impressions under $E_{\texttt{ads.txt}}$.

This surprising result has two explanations. First, the datasets used in these two studies were collected at different times. The *Inclusion* graph is built from the dataset which was collected in December 2015 (see § 4.1), whereas $E_{\texttt{ads.txt}}$ was isolated from the `ads.txt` dataset between January 2018 and April 2019. There is almost a three-year gap between these two datasets. And given how quickly the ad ecosystem is evolving (see § 6.3), the nodes in the *Inclusion* graph will be different. In this particular case, out of 1035 ad exchanges from the `ads.txt` dataset, only 128 are present in the *Inclusion* graph. Furthermore, out of those 128, only 19 were present in the manually selected 36 ad exchanges.

Second, the *Inclusion* graph is extremely dense (see § 6.3). Due to that, RTB Constrained simulation becomes insensitive to the number of ad exchanges after a certain number of ad exchanges have been selected in E. This is because a few well-connected ad exchanges can disperse the user impressions to the majority of the A&A domains in the *Inclusion* graph, and selecting more ad exchanges does not significantly change the amount of impressions learned. I demonstrate this behavior in § 6.4.3.

In the rest of the analysis, **I use $E_{\texttt{ads.txt}}$ as the list of ad exchanges for my RTB Constrained simulations**. Although, in the original publication of this study, I used 36 manually selected A&A domains as ad exchanges in my analysis, results from Figure 6.6 demonstrate that results under the two sets will be similar.

6.4.2.2 Node Blocking

To answer my third question, I must simulate the effect of "blocking" A&A domains on the *Inclusion* graph. A simulated user that blocks the A&A domain a_j will not make direct connections to it (the solid outlines in Figure 6.5). However, blocking a_j does **not** prevent a_j from tracking users indirectly: if the simulated user contacts ad exchange a_i, the impression may be forwarded to

a_j during the bidding process (the dashed outlines in Figure 6.5). For example, an extension that blocks a_2 in Figure 6.5 will prevent the user from seeing an ad, as well as prevent information flow to a_4 and a_5. However, blocking a_2 does not stop information from flowing to e_1, e_2, a_1, a_3, and even a_2!

I evaluate five different blocking strategies to compare their relative impact on user privacy under our three impression propagation models:

1. I randomly blocked 30% (310) of the A&A nodes from the *Inclusion* graph.[10]
2. I blocked the top 10% (103) of A&A nodes from the *Inclusion* graph, sorted by weighted PageRank.
3. I blocked all 594 A&A nodes from the Ghostery [73] blacklist.
4. I blocked all 412 A&A nodes from the Disconnect [52] blacklist.
5. I emulated the behavior of AdBlock Plus [7], which is a combination of whitelisting A&A nodes from the Acceptable Ads program [190], and blacklisting A&A nodes from EasyList [54]. After whitelisting, 634 A&A nodes are blocked.

I chose these methods to explore a range of graph theoretic and practical blocking strategies. Prior work has shown that the global connectivity of small-world graphs is resilient against random node removal [29], but I would like to empirically determine if this is true for ad network graphs as well. In contrast, prior work also shows that removing even a small fraction of top nodes from small-world graphs causes the graph to fracture into many subgraphs [132, 195]. Ghostery and Disconnect are two of the most widely-installed tracker blocking browser extensions, so evaluating their blacklists allows us to quantify how good they are at protecting users' privacy. Finally, AdBlock Plus is the most popular ad-blocking extension [122, 159], but contrary to its name, by default, it whitelists A&A companies that pay to be part of its Acceptable Ads program [8]. Thus, I seek to understand how effective AdBlock Plus is at protecting users' privacy under its default behavior.

6.4.3 Validation

To confirm that my simulations are representative of our ground-truth data, I perform some sanity checks. I simulate a single user in each model (who generates 5K impressions) and compare the resulting simulated inclusion trees to the original, real inclusion trees.

[10] I also randomly blocked 10% and 20% of A&A nodes, but the simulation results were very similar to that of random 30%.

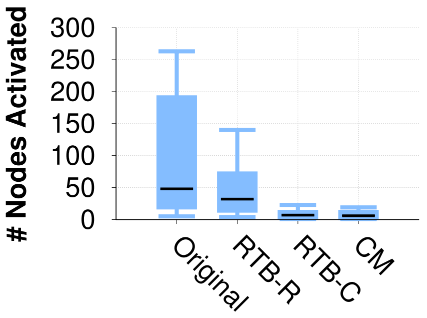


Figure 6.7: Number of nodes

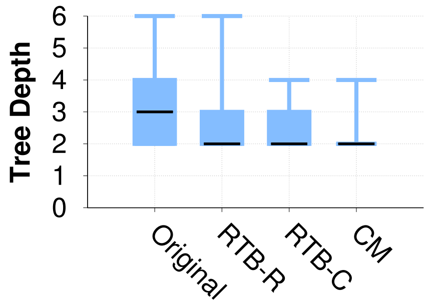


Figure 6.8: Tree depth

Figure 6.9: Comparison of the original and simulated inclusion trees. Each bar shows the 5^{th}, 25^{th}, 50^{th} (in black), 75^{th}, and 95^{th} percentile value.

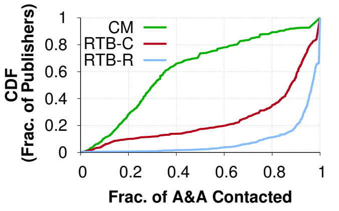


Figure 6.10: CDF of the fractions of A&A domains contacted by publishers in our original data that were **also** contacted in our three simulated models.

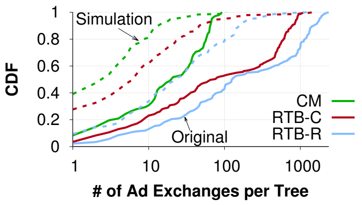


Figure 6.11: Number of ad exchanges in our original (solids lines) and simulated (dashed lines) inclusion trees.

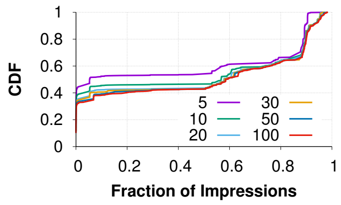


Figure 6.12: Fraction of impressions observed by A&A domains in RTB-C model when top x exchanges are selected.

First, I look at the number of nodes that are activated by direct propagation in trees rooted at each publisher. Figure 6.7 shows that out models are conservative in that they generate smaller trees: the median original tree contains 48 nodes, versus 32, seven, and six from our models. One caveat to this is that publishers in our simulated trees have a wider range of fan-outs than in the original trees. The median publishers in the original and simulated trees have 11 and 12 neighbors, respectively, but the 75^{th} percentile trees have 16 and 30 neighbors, respectively.

Second, I investigate the depth of the inclusion trees. As shown in Figure 6.8, the median tree depth in the original trees is three, versus two in all our models. The 75^{th} percentile tree depth in the original data is four, versus three in the RTB Relaxed and RTB Constrained models, and two in the most restrictive Cookie Matching-Only model. These results show that overall, my models are conservative in that they tend to generate slightly shorter inclusion trees than reality.

Third, I look at the set of A&A domains that are included in trees rooted at each publisher. For a publisher p that contacts a set A_p^o of A&A domains in my **o**riginal data, I calculate $f_p = |A_p^s \cap A_p^o|/|A_p^o|$, where A_p^s is the set of A&A domains contacted by p in **s**imulation. Figure 6.10 plots the CDF of f_p values for all publishers in our dataset, under our three models. I observe that for almost 80% publishers, 90% A&A domains contacted in the original trees are also contacted in trees generated by the RTB Relaxed model. This falls to 60% and 16% as the models become more restrictive.

Fourth, I examine the number of ad exchanges that appear in the original and simulated trees. Examining the ad exchanges is critical since they are responsible for all indirect dissemination of impressions. As shown in Figure 6.11, inclusion trees from our simulations contain an order of magnitude fewer ad exchanges than the original inclusion trees, regardless of model.[11] This suggests that indirect dissemination of impressions in my models will be conservative relative to reality.

Number of Selected Exchanges. Finally, I investigate the impact of exchanges in the RTB Constrained model. I select the top x A&A domains from $E_{\texttt{ads.txt}}$ sorted by the presence on the unique number of publishers' ads.txt files, then execute a simulation. As shown in Figure 6.12, with 20 or more exchanges the distribution of impressions observed by A&A domains stops growing, i.e., my RTB Constrained model is relatively insensitive to the number of exchanges. This is not surprising, given how dense the *Inclusion* graph is (see § 6.3). This also explains why we observed similar distributions under two different sets of ad exchanges in Figure 6.6.[12]

6.4.4 Results

I take our 200 simulated users and "play back" their browsing traces over the unmodified *Inclusion* graph, as well as graphs where nodes have been blocked using the strategies outlined above. I record the total number of impressions observed by each A&A domain, as well as the fraction of unique publishers observed by each A&A domain under different impression propagation models.

Triggered Edges. Table 6.3 shows the percentage of edges between A&A nodes that are triggered in the *Inclusion* graph under different combinations of impression propagation models and blocking strategies. No blocking/RTB Relaxed is the most permissive case; all other cases have

[11] Because each of my models assumes that a different set of A&A nodes are ad exchanges, we must perform three corresponding counts of ad exchanges in our original trees.

[12] In the original publications, I selected the top x A&A domains by out-degree to act as exchanges (subject to their in/out-degree ratio r being in the range $0.7 \leq r \leq 1.7$) and PageRank. Results were similar to Figure 6.12.

Table 6.3: Percentage of **E**dges that are triggered in the *Inclusion* graph during our simulations under different propagation models and blocking scenarios. I also show the percentage of edge **W**eights covered via triggered edges.

Blocking	**Cookie Matching-Only**		**RTB Constrained**		**RTB Relaxed**	
Scenarios	**%E**	**%W**	**%E**	**%W**	**%E**	**%W**
No Blocking	16.9	31.0	33.9	55.9	71.8	81.3
AdBlock Plus	12.3	28.0	25.6	50.3	48.4	68.6
Random 30%	12.1	21.8	22.1	34.2	48.7	54.8
Ghostery	3.52	9.87	6.82	18.2	13.5	21.9
Top 10%	6.03	5.01	8.18	5.52	26.8	13.4
Disconnect	2.98	3.66	4.72	6.01	16.3	11.6

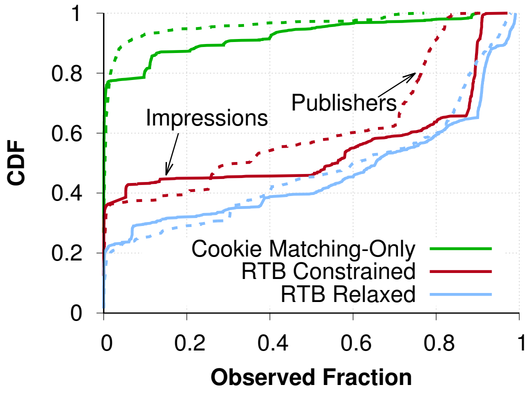


Figure 6.13: Fraction of impressions (solid lines) and publishers (dashed lines) observed by A&A domains under our three models, without any blocking.

fewer edges and weight because (1) the propagation model prevents specific A&A edges from being activated and/or (2) the blocking scenario explicitly removes nodes. Interestingly, AdBlock Plus fails to have a significant impact relative to the No Blocking baseline, in terms of removing edges or weight, under the Cookie Matching-Only and RTB Constrained models. Further, the top 10% blocking strategy removes fewer edges than Disconnect or Ghostery, but it reduces the remaining edge weight to roughly the same level as Disconnect, whereas Ghostery leaves more high-weight edges intact. These observations help to explain the outcomes of my simulations, which I discuss next.

No Blocking. First, I discuss the case where no A&A nodes are blocked in the graph. Figure 6.13 shows the fraction of total impressions (out of ∼5,300) and fraction of unique publishers (out of ∼190) observed by A&A domains under different propagation models. I find that the distribution of observed impressions under RTB Constrained is very similar to that of RTB Relaxed, whereas

Table 6.4: Top 10 nodes that observed the most impressions under my simulations with no blocking.

Cookie Matching-Only		**RTB Constrained**		**RTB Relaxed**	
doubleclick	90.1	google-analytics	97.1	pinterest	99.1
criteo	89.6	quantserve	92.0	doubleclick	99.1
quantserve	89.5	scorecardresearch	91.9	twitter	99.1
googlesyndication	89.0	youtube	91.8	googlesyndication	99.0
flashtalking	88.8	skimresources	91.6	scorecardresearch	99.0
mediaforge	88.8	twitter	91.3	moatads	99.0
adsrvr	88.6	pinterest	91.2	quantserve	99.0
dotomi	88.6	criteo	91.2	doubleverify	99.0
steelhousemedia	88.6	addthis	91.1	crwdcntrl	99.0
adroll	88.6	bluekai	91.1	adsrvr	99.0

observed impressions drop dramatically under the Cookie Matching-Only model. Specifically, the top 10% of A&A nodes in the *Inclusion* graph (sorted by impression count) observe more than 97% of the impressions in RTB Relaxed, 90% in RTB Constrained, and 29% in Cookie Matching-Only. I observe similar patterns for fractions of publishers observed across the three indirect propagation models. Recall that the Cookie Matching-Only and RTB Relaxed models function as lower- and upper-bounds on observability; that the results from the RTB Constrained model are so similar to the RTB Relaxed model is striking, given that only 128 nodes in the former spread impressions indirectly, versus 1,032 in the latter.

Although the overall fraction of observed impressions drops significantly in the Cookie Matching-Only model, Table 6.4 shows that the top 10 A&A domains observe 99%, 96%, and 89% of impressions on average under RTB Relaxed, RTB Constrained, and Cookie Matching-Only respectively. Some of the top-ranked nodes are expected, like DoubleClick, but other cases are more interesting. For example, Pinterest is connected to 178 publishers and 99 other A&A domains. In the Cookie Matching-Only model, it ranks 47 because it is directly embedded in relatively few publishers, but it ascends to rank seven and one, respectively, once indirect sharing is accounted for. This drives home the point that although Google is the most pervasively embedded advertiser around the web [33, 168], there are a roughly 52 other A&A domains that also observe greater than 91% of users' browsing behaviors (in the RTB Constrained model), due to their participation in major ad exchanges.

With Blocking. Next, I discuss the results when AdBlock Plus (i.e., the Acceptable Ads whitelist and EasyList blacklist) is used to block nodes. AdBlock Plus has essentially zero impact on the fraction of impressions observed by A&A domains: the results in Figure 6.14 under the RTB Con-

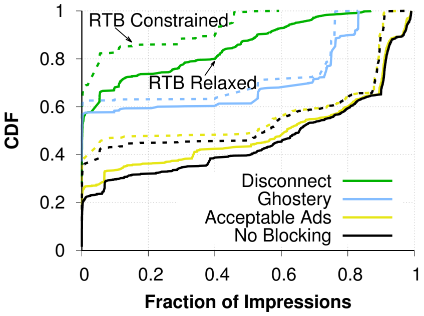


Figure 6.14: Disconnect, Ghostery, AdBlock Plus with Acceptable Ads.

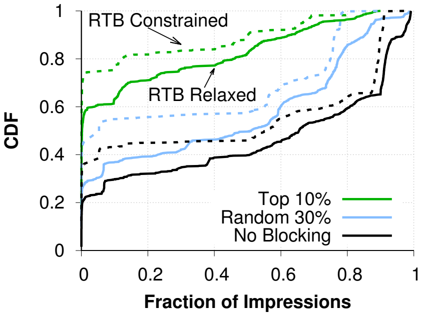


Figure 6.15: Top 10% and Random 30% of nodes.

Figure 6.16: Fraction of impressions observed by A&A domains under the **RTB Constrained** (dashed lines) and **RTB Relaxed** (solid lines) models, with various blocking strategies.

Table 6.5: Top 10 nodes that observed the most impressions in the **Cookie Matching-Only** and **RTB Constrained** models under various blocking scenarios. The numbers for the **RTB Relaxed** model (not shown) are slightly higher than those for RTB Constrained. Results under blocking random 30% nodes (not shown) are slighlty lower than no blocking.

AdBlock Plus w/ Acceptable Ads				Disconnect				Ghostery				Top 10 %			
CM-Only	**%**	**RTB Constrained**	**%**	**CM-Only**	**%**	**RTB Constrained**	**%**	**CM-Only**	**%**	**RTB Constrained**	**%**	**CM-Only**	**%**	**RTB Constrained**	**%**
doubleclick	90.0	google-analytics	97.0	amazonaws	43.7	amazonaws	59.3	criteo	75.0	google-analytics	83.1	rubiconproject	64.3	doubleclick	80.6
quantserve	89.5	youtube	91.7	3lift	41.5	revenuemantra	51.6	googlesyndication	74.7	youtube	77.4	amazon-adsystem	64.2	doubleverify	80.6
criteo	89.4	quantserve	91.6	zergnet	40.9	bidswitch	50.8	2mdn	74.5	betrad	76.2	googlesyndication	64.2	googlesyndication	80.6
googlesyndication	88.9	scorecardresearch	91.6	celtra	40.5	jwpltx	50.5	doubleclick	74.5	acexedge	76.2	mathtag	52.5	moatads	80.6
dotomi	88.6	skimresources	91.3	sonobi	40.4	basebanner	50.4	adnxs	73.3	vindicosuite	76.2	undertone	52.1	2mdn	80.6
flashtalking	88.6	twitter	91.1	bzgint	40.2	zergnet	46.0	adroll	73.3	2mdn	76.1	sitescout	50.1	twitter	80.6
adroll	88.5	pinterest	91.0	eyeviewads	40.2	sonobi	45.8	adsrvr	73.3	360yield	76.1	doubleclick	49.8	bluekai	80.6
adsrvr	88.5	addthis	90.9	simplereach	40.0	adnxs	45.8	adtechus	73.3	adadvisor	76.1	adtech	49.7	google-analytics	80.5
mediaforge	88.5	criteo	90.9	richmetrics	39.9	adsafeprotected	45.8	advertising	73.3	adap	76.1	adnxs	49.7	media	80.5
steelhousemedia	88.5	bluekai	90.8	kompasads	39.9	adsrvr	45.8	amazon-adsystem	73.3	adform	76.1	mediaforge	49.6	exelator	80.5

strained and RTB Relaxed models are almost coincident with those for the models when no blocking is applied at all. The problem is that the major ad networks and exchanges are all present in the Acceptable Ads whitelist, and thus all of their partners are also able to observe the impressions, even if they are (sometimes) prevented from actually showing ads to the user. Indeed, the top 10 nodes in Table 6.4 with no blocking and in Table 6.5 with AdBlock Plus are almost identical, save for some reordering.

Next, I examine Ghostery and Disconnect in Figure 6.14. As expected, the amount of information seen by A&A domains decreases when I block domains from these blacklists. Disconnect's blacklist [52] does a much better job of protecting users' privacy in our simulations: after blocking nodes using the Disconnect blacklist, 90% of the nodes see less than 40% of the impressions in the RTB Constrained model and less than 53% in the RTB Relaxed model. In contrast, when using the Ghostery blacklist [73], 90% of the nodes see less than 75% of the impressions in both RTB models. Table 6.5 shows that top 10 A&A domains are only able to observe at most 40–59% and 73–83% of

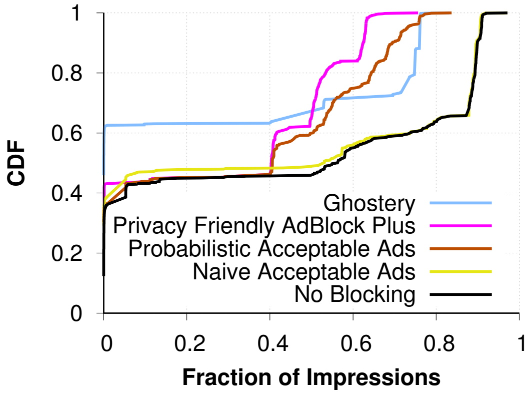


Figure 6.17: Fraction of impressions observed by A&A domains under the **RTB Constrained** with various blocking strategies. AdBlock Plus performance is divided into (naive) Acceptable Ads, Probabilistic Acceptable Ads, and Privacy Friendly criterias.

impressions when the Disconnect and Ghostery blacklists are used, respectively, depending on the indirect propagation model.

As shown in Figure 6.15, blocking the top 10% of A&A nodes from the *Inclusion* graph (sorted by weighted PageRank) causes almost as much reduction in observed impressions as Disconnect. Table 6.5 helps to orient the top 10% blocking strategy versus Disconnect and Ghostery in terms of the overall reduction in impression observability and the impact on specific A&A domains. In contrast, blocking 30% of the A&A nodes at random has more impact than AdBlock Plus, but less than Disconnect and Ghostery. Top 10 nodes under the “no blocking” and “random 30%” (not shown) strategies observe similar impression fractions. Both of these results agree with the theoretical expectations for small-world graphs, i.e., their connectivity is resilient against random blocking, but not necessarily targeted blocking.

I do not show results for our most restrictive model (i.e., Cookie Matching-Only) in Figure 6.16, since the majority of A&A domains view almost zero impressions. Specifically, 90% of A&A domains view less than 0.2%, 0.3%, and 11% of the impressions under Ghostery, Disconnect, and top 10% blocking. However, I do present the number of impressions seen by top 10 A&A domains in the Cookie Matching-Only model in Table 6.5, which shows that even under strict blocking strategies, top advertising companies still view 40–75% of the impressions.

Limitations in AdBlock Plus Blocking. I must translate rules from the EasyList and EasyPrivacy blacklists and the Acceptable Ads whitelist to use them in my simulations. Both of these lists include rules containing regular expressions, URLs, and even snippets of CSS; I simplify them to lists of effective 2^{nd}-level domains. Due to this translation, we may over-estimate impressions seen by the whitelisted A&A domains, and under-estimate impressions seen by blacklisted A&A domains. This could potentially explain the poor performance of AdBlock Plus blocking in my simulations. Therefore, to effectively simulate AdBlock Plus's behavior, we cannot simply extract 2^{nd}-level domains from blacklists.

To solve this issue, I generate a probability of each A&A being blocked at a given publisher p. In specific, I analyze all inclusion chains C_p associated with each publisher p. For each chain $c \in C_p$, I traverse c to check each resource URL against the blacklist rules. I simply keep count of the number of times a URL for a given A&A domain is blocked at p. Then, the probability of being blocked of an A&A domain at a given publisher p is simply $U_{BLOCKED}/U_{ALL}$, where U_{ALL} are all the URLs for a given A&A domain at p and $U_{BLOCKED} \in U_{ALL}$ are all the URLs which are blocked by AdBlock Plus. Then, I use these probabilities in my simulations to block a given A&A node at publisher p according to these probabilities.

Figure 6.17 shows the fraction of impressions observed under the RTB Constrained model with naive and probabilistic blocking using AdBlock Plus rules. The naive blocking is similar to that of shown in Figure 6.14, where EasyList and EasyPrivacy rules were simplified by extracting effective 2^{nd}-level domains. Probabilistic blocking is an improvement on the naive blocking using the methodology described above. “Ghostery" and “No Blocking" lines are shown for reference. We can see that AdBlock Plus performs much better under the probabilistic blocking scenario. However, it still under-performs Ghostery for 70% of A&A domains.

After the publication of this work, `eyeo` [63], the company which developed AdBlock Plus asked us to run our simulations on their “privacy-friendly" Acceptable Ads list. This additional list allows the user to view advertisements without third-party tracking [157]. **Note:** The “privacy-friendly" feature is not enabled by default when you install the extension, whereas the Acceptable Ads whitelisting option is. Figure 6.17 shows the results of RTB Constrained simulation results under the “privacy-friendly" feature. We do see an improvement in terms of impressions observed over the default behavior of AdBlock Plus, however, it still under-performs Ghostery for the majority of A&A domains.

Summary. Overall, there are three takeaways from these simulations. *First*, the “no blocking” simulation results show that top A&A domains are able to see the vast majority of users' browsing history, which is extremely troubling from a privacy perspective. For example, even under the most constrained propagation model (Cookie Matching-Only), DoubleClick still observes 90% of all im-

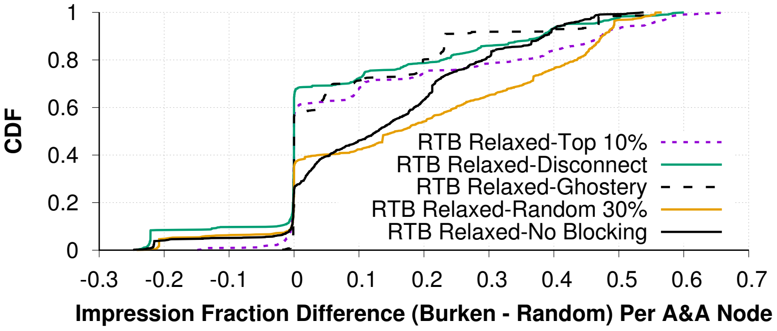


Figure 6.18: Difference of impression fractions observed by A&A nodes with simulations between Burklen et al. [31] and the random browsing model.

pressions generated by our simulated users. *Second*, it is troubling to observe that AdBlock Plus barely improves users' privacy, due to the Acceptable Ads whitelist containing high-degree ad exchanges. We do observe an improvement in AdBlock Plus when we block nodes probabilistically in our simulations, it still under-performs all other blocking strategies. *Third*, I find that users can improve their privacy by blocking A&A domains, but that the choice of blocking strategy is critically important. I find that the Disconnect blacklist offers the greatest reduction in observable impressions, while Ghostery offers significantly less protection. However, even when strong blocking is used, top A&A domains still observe anywhere from 40–80% of simulated users' impressions.

6.4.5 Random Browsing Model

Thus far, I have analyzed results for users that follow the browsing model from Burklen et al. [31]. This is, to the best of my knowledge, the only empirically validated browsing model.

To check the consistency of our simulation results, I ran additional simulations using a random browsing model, where the user chooses publishers purely at random and chooses whether to remain on a publisher or depart using a coin flip.

I plot the results of the random simulations in Figure 6.18 as the difference in the fraction of impressions observed by A&A domains under the RTB Relaxed model. Zero indicates that an A&A domain observed the same fraction of impressions in both the Burklen et al. and random user simulations, while <0 (>0) indicates that the node observed more impressions in the random (Burklen et al.) simulations. Between 20–60% of A&A nodes observe the same amount of impressions regardless of model, but this is because these nodes all observe **zero** impressions (i.e., they are blocked). This is why the fraction of A&A nodes that do not change between the browsing models is greatest with Disconnect. Although up to 10% of A&A nodes observe more impressions under the random browsing model, the majority of A&A nodes that observe at least one impression observe more overall under the Burklen et al. model.

Overall, Figure 6.18 demonstrates that the baseline browsing behavior exhibited by a user does have a significant impact on their visibility to A&A companies. For example, using the Burklen et al. model [31], the selected publishers contact top 10 A&A domains (sorted by PageRank) 2.6× more than those selected by the random browsing model (and 4.6× if we consider the top 10 A&A domains sorted by betweenness centrality).

Importantly, however, the relative effectiveness of blocking strategies remains the same under a random browsing model. Disconnect still performed the best, followed by top 10%, Ghostery, random 30%, and then AdBlock Plus. This suggests that my findings concerning the efficacy of blocking strategies generalize to users with different browsing behaviors.

6.5 Limitations

As with all simulated models, there are some limitations to this work.

First, my models of indirect impression dissemination are approximations. The Cookie Matching-Only and RTB Relaxed models should be viewed as lower- and upper-estimates, respectively, on the dissemination of impressions, not as accurate reflections of reality (for the reasons highlighted in Figure 6.5). I believe that the RTB Constrained model is a reasonable approximation, but even it has flaws: it may still exhibit false positives, if non-exchanges are included in the set of exchanges E, and false negatives if an actual exchange is not included in E. I described how data from the `ads.txt` study can be used to extract a list of ad exchanges and us it RTB Constrained simulations. The data for *Inclusion* graph and `ads.txt` was collected at different times. The ideal way to conduct this simulation would be to fetch the `ads.txt` file for the publisher p around the same time when inclusion resources are collected from p. Furthermore, it is not clear in general if ad exchanges always forward all impressions to all partners. For example, *private exchanges* that connect high-value publishers (e.g., The New York Times) to select pools of advertisers behave differently than their public cousins.

Second, these results are dependent on assumptions about the browsing behavior of users. I present results from two browsing models in § 6.4.5 and show that many of my headline results are robust. However, these findings should not be over-generalized: they are representative of an average user, yet specific individuals may experience different amounts of tracking.

Third, we must translate rules from the EasyList blacklist and the Acceptable Ads whitelist to use them in our simulations. Both of these lists include rules containing regular expressions, URLs, and even snippets of CSS; we simplify them to lists of effective 2^{nd}-level domains. Due to this translation, we may over-estimate impressions seen by the whitelisted A&A domains, and under-estimate impressions seen by blacklisted A&A domains. I present an improvement on this in § 6.4.4 by deriving probabilities of A&A domains being blocked at a given publisher, and using those probabilities during simulations. Note that the Ghostery and Disconnect blacklists are not affected by these issues.

Fourth, I analyze a dataset that was collected in December 2015. The structure of the *Inclusion* graph has almost certainly changed since then. Furthermore, the edge weights between nodes may differ depending on the initial set of publishers that are crawled. Although I demonstrate in § 6.2.4 that our dataset covers the vast majority of A&A domains, the connectivity, and weights between A&A domains may change over time, as ad campaigns and money shift.

Fifth, this dataset does not cover the mobile advertising ecosystem, which is known to differ from the web ecosystem [189]. Thus my results likely do not generalize to this area.

6.6 Sumamry

In this chapter, I introduced a novel graph model of the advertising ecosystem called an *Inclusion* graph. This representation is enabled by advances in browser instrumentation [17, 112] that allow researchers to capture the precise inclusion relationships between resources from different A&A domains [21]. Using a large crawled dataset from chapter 4, I show that the ad ecosystem is extremely dense. Furthermore, I compare our *Inclusion* graph representation to a *Referer* graph representation proposed by prior work [78], and show that the *Referer* graph has substantive structural differences that are caused by erroneously attributed edges.

I show that my proposed *Inclusion* graph can be used to implement empirically-driven simulations of the online ad ecosystem. My results demonstrate that under a variety of assumptions about user browsing and advertiser interaction behavior, top A&A domains observe the vast majority of users' browsing history. Even under realistic conditions where only a small number of well-connected ad exchanges indirectly share impressions, top 10% of A&A domains observe more than 90% impressions and 82% publishers.

I also evaluate a variety of ad and tracker blocking strategies in the context of my models, to understand their effectiveness at stopping A&A domains from learning users' browsing history. On one hand, I find that blocking the top 10% of A&A domains, as well as the Disconnect blacklist, does significantly reduce the observation of users' browsing. On the other hand, even these strategies still leak 40–80% of users' browsing history to top A&A domains, under realistic assumptions. This suggests that users who truly care about privacy on the web should adopt the most stringent blocking tools available, such as EasyList and EasyPrivacy, or consider disabling JavaScript by default with an extension like uMatrix [76].

Chapter 7

Conclusion

The rise of RTB has changed the privacy landscape significantly by forcing A&A companies to collaborate more closely with one another. Without exchanging user data with each other, A&A companies cannot successfully participate in RTB auctions. This massive amount of information-sharing has increased the privacy digital footprint of users; due to RTB, tracking data is not just observed by trackers embedded directly into web pages, but rather it is propagated to other A&A companies in the advertising ecosystem. This data dissemination is further exacerbated by the fact that ad exchanges send user impressions to several A&A companies to solicit bids during the RTB auction. This increased amount of information-sharing among A&A companies has given rise to the need for understanding the complexities and privacy implications of the modern ad ecosystem.

Although there has been prior empirical work on detecting information-sharing between A&A companies [5, 65, 147], these works have technical limitations which have prevented researchers from developing accurate models to demonstrate the privacy implications of RTB in the modern ad ecosystem. The primary limitation of prior works on detecting information-sharing is their reliance on heuristics that look for specific string signatures in HTTP messages. These heuristics are brittle in the face of obfuscation: for example, DoubleClick cryptographically hashes their cookies before sending them to other advertising partners [2, 147]. Additionally, analysis of *client-side* HTTP messages is insufficient to detect *server-side* information flows between A&A companies. This can happen if two ad networks decide to sync user tracking identifiers behind-the-scenes, without relying on redirect through a user's browser. These limitations may cause the privacy community to under-estimate the privacy digital footprint of users, which, in turn, may affect the development of effective privacy tools.

This thesis posits that RTB has increased collaboration among A&A companies, which, in turn, has increased privacy exposure for end-users. We need effective tools and methodologies to understand the privacy implications of RTB for users, to bridge the divide between the actual privacy landscape and our understanding of it. These techniques can provide a more realistic view of the online advertising ecosystem, and enable users to gain a more accurate view of their privacy digital footprint.

7.1 Contributions & Impact

In this thesis, I present methods and tools to understand the privacy implications of the modern ad ecosystem, taking into account RTB and information-sharing among A&A companies. In particular, my thesis makes the following contributions:

1. **Generic Methodology for Detecting Information-sharing Among A&A companies.** To address the limitations of existing techniques [5,65,147], I propose a novel methodology that can detect client- and server-side flows of information between arbitrary A&A companies using *retargeted ads*. Retargeted ads are the most specific form of behavioral advertisements, where a user is targeted with ads related to the exact products she has previously browsed.
   My key insight is to leverage retargeted ads as a mechanism for identifying information flows between arbitrary A&A companies. This methodology addresses the limitations of prior work because it relies on the *semantics* of how exchanges serve ads, rather than focusing on specific cookie matching *mechanisms*. Specifically, instead of relying on HTTP messages to detect cookie matching, it relies on causality. Thus, this methodology can defeat obfuscation and can detect server-side information sharing.
   Based on extensive experiments, I demonstrate that information-sharing among A&A companies can be divided in to four categories that reveal 1) the pair of A&A companies that shared information to serve the retargeted ad, and 2) the mechanism they used to share the data (e.g., cookie matching, server-side matching). Although I confirm that the key information-sharing mechanism is client-side *cookie-matching*, my proposed methodology successfully identifies server-side matching flows between Google services.
2. **Identification of Ad Exchanges via `ads.txt` Standard.** To model users' privacy digital footprint accurately, we need to identify not only the information-sharing relationships among A&A domains but also a list of A&A domains that function as ad exchanges. Identifying ad

exchanges accurately is crucial since they disperse user impressions to multiple other A&A companies to solicit bids.

I identify a ground-truth set of ad exchanges by conducting a longitudinal analysis of a transparency standard called `ads.txt` [83], which was introduced to combat ad fraud by helping ad buyers verify authorized digital ad sellers. `ads.txt` is meant to bring more transparency to the opaque ecosystem of RTB, by making it explicit which third-party domains in a given first-party context are ad exchanges. I use this as an opportunity to gather a list of ad exchanges involved in the RTB ecosystem.

In particular, I conduct a 15-months longitudinal study of the standard to gather a list of A&A domains that are labeled as ad exchanges (authorized sellers) by publishers in their `ads.txt` files. Through my analysis on Alexa Top-100K, I observed that over 60% of the publishers who run RTB ads have adopted the `ads.txt` standard. This widespread adoption allowed me to explicitly identify over 1,000 A&A domains domains belonging to ad exchanges.

3. **Modeling User's Digital Privacy Footprint.** Using the information flows between A&A companies and the list of exchanges, I model the advertising ecosystem in the form of a graph called an *Inclusion* graph. By simulating browsing traces for 200 users based on empirical data, I show that the *Inclusion* graph can be used to model the diffusion of user tracking data across the advertising ecosystem.

   Through my analysis, I demonstrate that due to RTB, the majority of A&A domains observe the vast majority of users' browsing history. Even under restrictive conditions, where only a small number of well-connected ad exchanges indirectly share impressions during RTB auctions, the top 10% of A&A domains observe more than 91% of impressions and 82% of visited publishers. This is a key result as it highlights that A&A domains observe far greater amounts of user information than what has been demonstrated by prior works [5, 61].

   I also evaluate the effectiveness of privacy tools (e.g., AdBlock Plus, Disconnect) at protecting users' privacy in the presence of RTB. I find that AdBlock Plus (the world's most popular ad-blocking browser extension [122, 159]) is ineffective at protecting users' privacy because major ad exchanges are whitelisted under the Acceptable Ads program [190]. In contrast, Disconnect [52] blocks the most information flows to advertising domains, followed by the removal of top 10% A&A domains. However, the most important observation throughout these experiments is that even with strong blocking methods, major A&A domains still observe 40–70% of user impressions.

The work described in this thesis has been published at top-level security and privacy venues. My work on modeling the users' privacy digital footprint received the best student paper at the *Future of Privacy Forum's Annual Privacy Papers for Policymakers Awards* [144]. This venue bridges the gap between academia and policymakers by summarizing and distributing relevant papers directly to US lawmakers and their staff.

All the data from my thesis is made public for the community's benefit. Datasets can be found at:

1. https://personalization.ccs.neu.edu/Projects/Retargeting/
2. https://personalization.ccs.neu.edu/Projects/Adstxt/
3. https://personalization.ccs.neu.edu/Projects/AdGraphs/

Additionally, the tool used to crawl inclusion chains in my thesis is also publicly available at https://github.com/sajjadium/DeepCrawling.

7.2 Limitations

In this section, I describe the limitations that should be considered for the results presented in this thesis.

First, the dataset for information-sharing was collected in December 2015. This means that while the methodology proposed in this thesis can be used for future studies, researchers should be cautious when attempting to generalize specific results from my work. This is because of the fast-evolving nature of the advertising ecosystem, in which A&A companies appear, merge, go out of business, and adopt new technologies over time.

Second, in this thesis, I only consider static, image-based advertisements served through RTB auctions. Video ads served through RTB are on the rise and should be integrated into the experiments for future studies [15, 163, 196]. This way, we can also capture those A&A companies that specialize in serving video ads.

Third, I have studied the information-sharing relationships on the web. However, my results may not generalize to the mobile advertising ecosystem, since it is known to differ from the web ecosystem [189].

Fourth, I rely on EasyList [54] and EasyPrivacy [55] to detect inclusion chains that end up serving advertisements. These lists are manually curated over time and may have false negatives. Furthermore, in my analysis I consider effective 2^{nd}-level domains (e.g., google.com will be `google`), whereas these lists include rules containing regular expressions, URLs, and even snippets of CSS. I simplify these rules to lists of effective 2^{nd}-level domains for my comparison, which could introduce some errors into my analysis of the capabilities of software relying on these lists (see § 6.5). In § 6.4.4, I provide a way to better model these rules.

Finally, to completely understand how much information is learned by an A&A company, we need to account for all the domains it owns. For example, Google owns many A&A related domains like DoubleClick, Google Analytics, Google Tag Manager, *etc.*. Although in my analysis I cluster domains based on the parent company, this clustering is done manually. A better approach for clustering might be to use external tools like WhoTracksMe [192].

Although my thesis addresses the technical limitations of prior work to provide much better estimates of users' privacy, we may still be under-estimating the digital footprint of users because of the above-mentioned limitations. The results provided in this thesis are from a specific snapshot, and might not capture privacy leakage to A&A companies that were inactive during the crawls. Longitudinal analysis is important to get a better understanding of users' privacy health. Additionally, this thesis does not account for A&A companies that specialize in specific business models (e.g., video ads, email-based ads [60]). Another reason why these results under-estimate privacy leakage is that I do not account for (offline) information-sharing with data brokers [24] and information aggregation across multiple devices (i.e., cross-device tracking) [201].

7.3 Lessons Learned

In this section, I share the key lessons from this thesis and discuss the topics which need attention from the privacy community.

The ad ecosystem is extremely complex and opaque, and it is important to identify the right methods and tools to study a certain problem. For example, if the goal is to understand the prevalence of trackers, there are public tools like OpenWPM [59], which the community can use. However, if the goal is to understand information flows between A&A companies, a tool like the one used in this thesis, which provides detailed inclusion logs, is critical. Researchers working in this space should strive to adopt the most appropriate tool for their particular use case.

The importance of manual analysis cannot be understated. The data collected through large-scale crawls can be overwhelming, and coupled with the complex nature of the web, analysis can often become a daunting task. One thing which always helped me was to randomly down-sample logs and look at them manually. Investing a few hours initially on this process not only helped me understand how information flows on the web, but more often than not gave me key insights that I would have missed otherwise.

I believe that researchers should shift their focus towards other sources of privacy leakage. Over the years, significant progress towards unscrambling the complex ad ecosystem has been made. From understanding the prevalence of trackers to the specific mechanisms of tracking (e.g., cookies, fingerprinting), there is a plethora of literature available. In this thesis, I improve upon prior work to provide techniques and methods for detecting flows of tracking information between arbitrary A&A companies. Although we should continue to replicate existing studies to understand the fast-evolving nature of the ad ecosystem, there are important topics that need the attention of the privacy community.

There are three key areas that I think should be focused on. First, besides some initial work on cross-device tracking [201], we still lack a good understanding of which A&A companies track users across multiple devices, what mechanisms they use, and whether they share that information with other business partners. Second, A&A companies are gradually moving towards acquiring user information from data brokers like Acxiom and Bluekai. While we know that data brokers like Acxiom and Bluekai exist and they have business relationships with several A&A companies, we don't have a complete understanding of the extent to which data sharing occurs through data brokers and Data Management Platforms (DMPs). Third, we need tools to audit compliance with privacy initiatives (e.g., GDPR, EU cookie directive) by A&A companies. These initiatives are against the business interests of the ad industry, and A&A companies try their best to resist them [185]. The research community is well positioned to provide tools and insights that promote accountability in this sector.

With the amount of information users put online and the hunger of A&A companies to turn this information into profit, the privacy landscape is getting worse. Although there are privacy tools (e.g., tracker and ad-blocking extensions) that decrease users' digital footprint, in chapter 6, I show that these tools are not as effective as we think they are. Blocking individual trackers is not enough as user data is transferred to the blocked trackers through extensive information-sharing. What is even more troubling is that there are instances where sensitive private information is leaked and sometimes even stolen, and there is nothing users can do about it [32, 126]. At a minimum, users need access to transparency tools that enable them to understand and manage the data about them held by third-parties.

As of this writing, there are only a handful of companies that provide such transparency tools to let users control their data [24], and they are prone to have issues. In particular, they lack coverage [14, 194], exclude sensitive user attributes [48], and infer noisy and irrelevant interests [24, 50, 186].

Even with these bleak conditions for privacy, users can still take certain steps to minimize their privacy digital footprint. In particular, they can take the following steps:

- Make themselves more aware of the privacy issues.
- Use a privacy-friendly browser like Firefox, Brave, or Safari.
- Use effective privacy protection tools like uBlock Origin [90] or uMatrix [76].
- Use opt-out services provided by the Network Advertising Initiative [97] and the Digital Advertising Alliance [12].

Regulators can also play a part by introducing more privacy legislation like the GDPR [41] and the California Consumer Privacy Act [70].

7.3.1 Future Directions

While my thesis makes significant contributions towards the understanding of information-sharing among A&A companies, and, in turn, the development of privacy-enhancing tools, there is much more work needed to be done.

First, the ultimate goal of my research is not just to measure information flows among A&A companies, but to facilitate the development of privacy-enhancing tools that can help users be informed and in control of their private data. To that end, using my methodology to detect information-sharing and then using my information diffusion models, researchers can build tools to let users know which A&A company is viewing how much of their information. Furthermore, from the analysis I provide on evaluating the efficacy of tracker and ad-blocking extensions, users can then be provided estimates on how their information leakage will change if they block certain A&A nodes in the advertising graph.

Second, my content- and platform-agnostic methodology can be used to study the flows of information across multiple platforms and devices. To gain a complete picture of user behavior and interests across all devices, A&A companies identify all devices associated with a particular user through cross-device tracking [30, 201]. While prior works highlight that cross-device tracking exists, they do not tell us which attribute(s) (e.g., email address, username, advertising ID) facilitated cross-device tracking, nor do they identify the A&A companies which share information across multiple devices.

The detection of information-sharing flows between A&A companies across devices is particularly challenging since the tracking mechanisms on these devices differ. For example, cookies are used to track users on desktop devices, while advertising ID is used to track users on mobile apps. My proposed methodology can be used to study cross-device tracking since it does not look for patterns or identifiers in the network traffic, but rather relies on the causal inference of how a retargeted ad is shown.

Third, my work can be used for auditing and informing future privacy policies. Recently, browser vendors like Firefox, Safari, and Brave have started taking privacy initiatives. Firefox removed cross-site tracking with their version 65 [66] and is taking measures to combat website fingerprinting [67], and third-party storage access for trackers [68]. Safari has taken similar steps to prevent cross-site tracking by introducing `Intelligent Tracking Prevention` [193]. Brave, on the other hand, provides built-in tracker and ad blocking [28].

While these initiatives are positive steps towards greater privacy for users, we as researchers need to make sure that browsers are delivering what they are promising. We also have an opportunity to work with browser vendors to help them address any short-comings in their approaches. For example, in the past, companies have been known to track users by working around the privacy policies established by browsers [44].

The ideal way to conduct such studies would be to perform large-scale crawls on these specific browsers and collect detailed inclusion logs for further analysis. These logs can then be inspected for unwanted inclusions and information-sharing. Persona-based experiments, as described in § 4.1.3, can be used to gather concrete evidence for information-sharing. However, unlike Chrome, which provides detailed logs through Chrome Debugging Protocol [38], development tools provided by other browsers are limited in scope and could require extensive instrumentation.

Another important privacy initiative is the General Data Protection Regulation (GDPR) [41], which went into force in May 2018 by the European Commission for EU users. Under GDPR, both first- and third-parties need to obtain informed consent regarding the collection and processing of data. Regulatory authorities can use methods and techniques proposed in this thesis to analyze the flows of information between third parties, to study the relationships between user consent and information flows between third-parties. Regulators can see not only information flows to trackers present on websites, but also information-sharing behind the scenes (e.g., through or because of RTB).

Finally, my work can be extended to study other standards that require information-sharing among A&A companies. For example, Header Bidding (HB) is an emerging programmatic advertising mechanism that aims to remove the middle-man (ad exchanges) from the bidding process [51]. The key difference between RTB and HB is that under RTB, ad exchanges solicit bids and the client only observes the winning DSP. Whereas, under HB, the publisher chooses which DSPs can bid on its inventory and can receive bid responses from all the participating DSPs. Researchers can use the lessons from this thesis to evaluate information leakage under the HB model.

Bibliography

[1] Share of real-time bidding in digital display advertising spending in the united states from 2012 to 2018. Statista, March 2014. https://www.statista.com/statistics/267762/share-of-rtb-in-digital-display-ad-spend-in-the-us/.

[2] Real-time bidding protocol, February 2016. https://developers.google.com/ad-exchange/rtb/cookie-guide.

[3] U.s ad spending: The emarketer forecast for 2017. eMarketer, March 2017. https://www.emarketer.com/Report/US-Ad-Spending-eMarketer-Forecast-2017/2001998.

[4] Worldwide analysis on the real-time bidding market, 2019 to 2024 - anticipated to record a cagr of 32.9 CISON PR Newswire, March 2019. https://www.prnewswire.com/news-releases/worldwide-analysis-on-the-real-time-bidding-market-2019-to-2024---antici
html.

[5] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proc. of CCS*, 2014.

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

[7] Adblock plus: Surf the web without annoying ads! eyeo GmbH. https://adblockplus.org.

[8] Allowing acceptable ads in adblock plus. eyeo GmbH. https://adblockplus.org/acceptable-ads.

[9] Simplify your ads.txt management. Ads.txt Guru, 2018. https://adstxt.guru/publishers/.

[10] Ads.txt Manager | Free | Easily Manage Ads.txt Files. Ads.txt Manager. https://www.adstxtmanager.com.

[11] L. Agarwal, N. Shrivastava, S. Jaiswal, and S. Panjwani. Do not embarrass: Re-examining user concerns for online tracking and advertising. In *Proc. of the Workshop on Usable Security*, 2013.

[12] D. A. Alliance. Webchoices browser check. http://optout.aboutads.info.

[13] H. Almeida, D. Guedes, W. Meira, and M. J. Zaki. Is there a best quality metric for graph clusters? In *Proc. of ECML PKDD*, 2011.

[14] A. Andreou, G. Venkatadri, O. Goga, K. P. Gummadi, P. Loiseau, and A. Mislove. Investigating Ad Transparency Mechanisms in Social Media: A Case Study of Facebook's Explanations. In *Proc of NDSS*, 2018.

[15] AppNexus. Appnexus sees steep rise in video business, with 230% growth of rtb video spend, Feb. 2018. https://www.appnexus.com/company/pressroom/appnexus-sees-steep-rise-in-video-business-with-230-growth-of-rtb-video-

[16] Appnexus enforces ads.txt in broader push for industry transparency. AppNexus, Feb. 2018. https://www.appnexus.com/company/pressroom/appnexus-enforces-adstxt-in-broader-push-for-industry-transparency.

[17] S. Arshad, A. Kharraz, and W. Robertson. Include me out: In-browser detection of malicious third-party content inclusions. In *Proc. of Intl. Conf. on Financial Cryptography*, 2016.

[18] M. Ayenson, D. J. Wambach, A. Soltani, N. Good, and C. J. Hoofnagle. Flash cookies and privacy ii: Now with html5 and etag respawning. *Available at SSRN 1898390*, 2011.

[19] R. Balebako, P. G. Leon, R. Shay, B. Ur, Y. Wang, and L. F. Cranor. Measuring the effectiveness of privacy tools for limiting behavioral advertising. In *Proc. of W2SP*, 2012.

[20] P. Barford, I. Canadi, D. Krushevskaja, Q. Ma, and S. Muthukrishnan. Adscape: Harvesting and analyzing online display ads. In *Proc. of WWW*, 2014.

[21] M. A. Bashir, S. Arshad, , W. Robertson, and C. Wilson. Tracing information flows between ad exchanges using retargeted ads. In *Proc. of USENIX Security Symposium*, 2016.

[22] M. A. Bashir, S. Arshad, E. Kirda, W. Robertson, and C. Wilson. How Tracking Companies Circumvented Ad Blockers Using WebSockets. In *Proceedings of the Internet Measurement Conference (IMC 2018)*, Boston, MA, October 2018.

[23] M. A. Bashir, S. Arshad, and C. Wilson. "Recommended For You": A First Look at Content Recommendation Networks. In *Proc. of IMC*, 2016.

[24] M. A. Bashir, U. Farooq, M. Shahid, M. F. Zaffar, and C. Wilson. Quantity vs. Quality: Evaluating User Interest Profiles Using Ad Preference Managers. In *Proc of NDSS*, 2019.

[25] M. A. Bashir and C. Wilson. Diffusion of User Tracking Data in the Online Advertising Ecosystem. In *Proc. of PETS*, 2018.

[26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), 2008.

[27] T. Book and D. S. Wallach. A case of collusion: A study of the interface between ad libraries and their apps. In *Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones*, SPSM '13, 2013.

[28] Brave. The browser that rethinks the web. https://brave.com/features/.

[29] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web: Experiments and models. In *Proc. of WWW*, 2000.

[30] J. Brookman, P. Rouge, A. Alva, and C. Yeung. Cross-device tracking: Measurement and disclosures. In *Proc. of PETS*, 2017.

[31] S. Burklen, P. J. Marron, S. Fritsch, and K. Rothermel. User centric walk: An integrated approach for modeling the browsing behavior of users on the web. In *Annual Symposium on Simulation*, April 2005.

[32] C. Cadwalladr and E. Graham-Harrison. Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach,

Mar. 2018. https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election.

[33] A. Cahn, S. Alfeld, P. Barford, and S. Muthukrishnan. An empirical study of web cookies. In *Proc. of WWW*, 2016.

[34] J. M. Carrascosa, J. Mikians, R. Cuevas, V. Erramilli, and N. Laoutaris. I always feel like somebody's watching me: Measuring online behavioural advertising. In *Proc. of ACM CoNEXT*, 2015.

[35] C. Castelluccia, M.-A. Kaafar, and M.-D. Tran. Betrayed by your ads!: Reconstructing user profiles from targeted ads. In *Proc. of PETS*, 2012.

[36] F. Chanchary and S. Chiasson. User perceptions of sharing, advertising, and tracking. In *Proc. of the Workshop on Usable Security*, 2015.

[37] Y. Chen. Domain spoofing remains a huge threat to programmatic. Digiday, Feb. 2017. https://digiday.com/marketing/domain-spoofing-remains-an-ad-fraud-problem/.

[38] Chrome devtools protocol viewer. GitHub. https://developer.chrome.com/devtools/docs/debugger-protocol.

[39] Clickbot.A User Agent String. Distil Networks), Nov. 2016. https://www.distilnetworks.com/bot-directory/bot/clickbot-a/.

[40] Cliqz - the no-compromise browser. Cliqz GmbH. https://cliqz.com/en/.

[41] E. Commission. Data protection in the eu. https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en.

[42] Standard banner sizes list. Bannersnack Blog, Nov. 2008. https://blog.bannersnack.com/banner-standard-sizes/.

[43] Guide to ad sizes. Google. https://support.google.com/adsense/answer/6002621?hl=en.

[44] S. Cowley and J. Pepitone. Google to pay record $22.5 million fine for Safari privacy evasion. CNNMoney, Aug. 2012. http://money.cnn.com/2012/08/09/technology/google-safari-settle/index.html.

[45] Criteo ranking by Econsultancy. http://www.criteo.com/resources/e-consultancy-display-retargeting-buyers-guide/.

[46] N. Daswani, C. Mysen, V. Rao, and S. Weis. Online advertising fraud. *Crimeware Underst. New Attacks Defenses*, 40, 01 2008.

[47] N. Daswani, T. G. C. Quality, S. Teams, and G. Inc. The anatomy of clickbot.a. In *USENIX Hotbots*, 2007.

[48] A. Datta, M. C. Tschantz, and A. Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. In *Proc. of PETS*, 2015.

[49] V. Dave, S. Guha, and Y. Zhang. Measuring and fingerprinting click-spam in ad networks. In *Proc. of SIGCOMM*, 2012.

[50] M. Degeling and J. Nierhoff. Tracking and tricking a profiler: Automated measuring and influencing of bluekai's interest profiling. In *Proc. of WPES*, 2018.

[51] A. Dey. Header Bidding vs RTB: Understanding the Differences. Blognife, Sept. 2018. https://blognife.com/2018/09/08/header-bidding-vs-rtb-understanding-the-differences/.

[52] Disconnect defends the digital you. Disconnect Inc. https://disconnect.me/.

[53] C. Dolin, B. Weinshel, S. Shan, C. M. Hahn, E. Choi, M. L. Mazurek, and B. Ur. Unpacking Perceptions of Data-Driven Inferences Underlying Online Targeting and Personalization. In *Proc. of CHI*, 2018.

[54] Easylist. The EasyList authors. https://easylist.to/easylist/easylist.txt.

[55] Easyprivacy. The EasyList authors. https://easylist.to/easylist/easyprivacy.txt.

[56] P. Eckersley. How unique is your web browser? In *Proc. of PETS*, 2010.

[57] M. Egele, C. Kruegel, E. Kirda, and G. Vigna. Pios: Detecting privacy leaks in ios applications. In *Proc of NDSS*, 2011.

[58] US Programmatic Ad Spending Forecast Update 2018. eMarketer, Oct. 2018. https://www.emarketer.com/content/us-programmatic-ad-spending-forecast-update-2018.

[59] S. Englehardt. Openwpm. https://github.com/mozilla/OpenWPM.

[60] S. Englehardt, J. Han, and A. Narayanan. I never signed up for this! privacy implications of email tracking. *PoPETs*, 2018(1):109–126, 2018.

[61] S. Englehardt and A. Narayanan. Online tracking: A 1-million-site measurement and analysis. In *Proc. of CCS*, 2016.

[62] S. Englehardt, D. Reisman, C. Eubank, P. Zimmerman, J. Mayer, A. Narayanan, and E. W. Felten. Cookies that give you away: The surveillance implications of web tracking. In *Proc. of WWW*, 2015.

[63] Eyeo. Putting you in charge of a fair, profitable web. https://eyeo.com/, Accessed on 07/12/19.

[64] M. Falahrastegar, H. Haddadi, S. Uhlig, and R. Mortier. The rise of panopticons: Examining region-specific third-party web tracking. In *Proc of. Traffic Monitoring and Analysis*, 2014.

[65] M. Falahrastegar, H. Haddadi, S. Uhlig, and R. Mortier. Tracking personal identifiers across the web. In *Proc. of PAM*, 2016.

[66] Firefox. Changing our approach to anti-tracking, August 2018. https://blog.mozilla.org/futurereleases/2018/08/30/changing-our-approach-to-anti-tracking/.

[67] Firefox. Protections against fingerprinting and cryptocurrency mining available in firefox nightly and beta, April 2019. https://blog.mozilla.org/futurereleases/2019/04/09/protections-against-fingerprinting-and-cryptocurrency-mining-available-i

[68] Firefox. What does the storage access policy block?, June 2019. https://developer.mozilla.org/en-US/docs/Mozilla/Firefox/Privacy/Storage_access_policy#What_does_the_storage_access_policy_block.

[69] Ads.txt Industry Dashboard. firstimpression.io, May 2019. https://adstxt.firstimpression.io/.

[70] C. for Consumer Privacy. California consumer privacy act. https://www.caprivacy.org.

[71] G. Franken, T. V. Goethem, and W. Joosen. Who left open the cookie jar? a comprehensive evaluation of third-party cookie policies. In *Proc. of USENIX Security Symposium*, 2018.

[72] A. Ghosh, M. Mahdian, P. McAfee, and S. Vassilvitskii. To match or not to match: Economics of cookie matching in online advertising. In *Proc. of EC*, 2012.

[73] Ghostery: faster, cleaner, and safer browsing. Cliqz International GmbH i.Gr. https://www.ghostery.com/.

[74] P. Gill, V. Erramilli, A. Chaintreau, B. Krishnamurthy, K. Papagiannaki, and P. Rodriguez. Follow the money: Understanding economics of online aggregation and advertising. In *Proc. of IMC*, 2013.

[75] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[76] GitHub. umatrix: Point and click matrix to filter net requests according to source, destination and type., October 2014. https://github.com/gorhill/uMatrix.

[77] R. Gomer, E. M. Rodrigues, N. Milic-Frayling, and M. C. Schraefel. Network analysis of third party tracking: User exposure to tracking cookies through search. In *Proc. of IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 2013.

[78] R. Gomer, E. M. Rodrigues, N. Milic-Frayling, and M. C. Schraefel. Network analysis of third party tracking: User exposure to tracking cookies through search. In *Prof. of IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 2013.

[79] Google. Google Privacy & Terms. Google Inc. https://policies.google.com/privacy?hl=en-US#footnote-combine-info, Accessed on 06/25/19.

[80] Declare authorized sellers with ads.txt. Google. https://support.google.com/admanager/answer/7441288?hl=en.

[81] Google strengthens ads.txt enforcement. Ad Exchanger, July 2018. https://adexchanger.com/ad-exchange-news/google-strengthens-ads-txt-enforcement/.

[82] M. C. Grace, W. Zhou, X. Jiang, and A.-R. Sadeghi. Unsafe exposure analysis of mobile in-app advertisements. In *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WISEC '12, 2012.

[83] O. W. Group. IAB Tech Lab ads.txt Specification Version 1.0.2. IAB Tech Lab, Mar. 2019. https://iabtechlab.com/wp-content/uploads/2019/03/IAB-OpenRTB-Ads.txt-Public-Spec-1.0.2.pdf.

[84] O. W. Group. IAB Tech Lab Authorized Sellers for Apps (app-ads.txt) Version 1.0. IAB Tech Lab, Mar. 2019. https://iabtechlab.com/wp-content/uploads/2019/03/app-ads.txt-v1.0-final-.pdf.

[85] O. W. Group. IAB Tech Lab Sellers.json DRAFT FOR PUBLIC COMMENT v1.0. IAB Tech Lab, Apr. 2019. https://iabtechlab.com/wp-content/uploads/2019/04/Sellers.json-Public-Comment-April-11-2019.pdf.

[86] S. Guha, B. Cheng, and P. Francis. Challenges in measuring online advertising systems. In *Proc. of IMC*, 2010.

[87] H. Haddadi. Fighting online click-fraud using bluff ads. *SIGCOMM Comput. Commun. Rev.*, 40(2):21–25, Apr. 2010.

[88] A. Hannak, P. Sapieżyński, A. M. Kakhki, B. Krishnamurthy, D. Lazer, A. Mislove, and C. Wilson. Measuring Personalization of Web Search. In *Proc. of WWW*, 2013.

[89] J. Hercher. Google Strengthens Ads.txt Enforcement. ad exchanger, July 2018. https://adexchanger.com/ad-exchange-news/google-strengthens-ads-txt-enforcement/.

[90] R. Hill. ublock origin - an efficient blocker for chromium and firefox. fast and lean. https://github.com/gorhill/uBlock.

[91] R. Hill. ws-gateway websocket circumvention? #1936. GitHub, Aug. 2016. https://github.com/gorhill/uBlock/issues/1936.

[92] D. Howell. How to protect your privacy and remove data from online services. Tech Radar, January 2015. http://www.techradar.com/news/internet/how-to-protect-your-privacy-and-remove-data-from-online-services-1291515

[93] M. D. Humphries and K. Gurney. Network 'small-world-ness': A quantitative method for determining canonical network equivalence. *PLoS One*, 3(4), 2008.

[94] How Adform Discovered HyphBot. AdForm, Nov. 2017. https://site.adform.com/media/85132/hyphbot_whitepaper_.pdf.

[95] What is an untrustworthy supply chain costing the US digital advertising industry? Interactive Advertising Bureau (IAB), Nov. 2015. https://www.iab.com/wp-content/uploads/2015/11/IAB_EY_Report.pdf.

[96] M. Ikram, H. J. Asghar, M. A. Kâafar, B. Krishnamurthy, and A. Mahanti. Towards seamless tracking-free web: Improved detection of trackers via one-class learning. *PoPETs*, 2017(1):79–99, 2017.

[97] N. A. Initiative. Opt out of interest-based advertising. http://optout.networkadvertising.org.

[98] The four types of domain spoofing. Integral Ads, Feb. 2015. https://insider.integralads.com/the-four-types-of-domain-spoofing/.

[99] U. Iqbal, Z. Shafiq, and Z. Qian. The ad wars: Retrospective measurement and analysis of anti-adblock filter lists. In *Proc. of IMC*, 2017.

[100] U. Iqbal, P. Snyder, S. Zhu, B. Livshits, Z. Qian, and Z. Shafiq. Adgraph: A graph-based approach to ad and tracker blocking. In *Proc. of IEEE Symposium on Security and Privacy*, May 2020.

[101] V. Jatain. What is Domain Spoofing? Ad PushUp, Nov. 2019. https://www.adpushup.com/blog/what-is-domain-spoofing/.

[102] V. Kalavri, J. Blackburn, M. Varvello, and K. Papagiannaki. Like a pack of wolves: Community structure of web trackers. In *Proc. of Passive and Active Measurement*, 2016.

[103] S. Kamkar. Evercookie - virtually irrevocable persistent cookies., September 2010. http://samy.pl/evercookie/.

[104] T. Kohno, A. Broido, and K. Claffy. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2(2):93–108, 2005.

[105] M. Koster. A Standard for Robot Exclusion, 2007. http://www.robotstxt.org/orig.html.

[106] B. Krishnamurthy, D. Malandrino, and C. E. Wills. Measuring privacy loss and the impact of privacy protection in web browsing. In *Proc. of the Workshop on Usable Security*, 2007.

[107] B. Krishnamurthy, K. Naryshkin, and C. Wills. Privacy diffusion on the web: A longitudinal perspective. In *Proc. of WWW*, 2009.

[108] B. Krishnamurthy and C. Wills. Privacy leakage vs. protection measures: the growing disconnect. In *Proc. of W2SP*, 2011.

[109] B. Krishnamurthy and C. E. Wills. Generating a privacy footprint on the internet. In *Proc. of IMC*, 2006.

[110] B. Krishnamurthy and C. E. Wills. On the leakage of personally identifiable information via online social networks. 2009.

[111] J. Larisch, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. CRLite: a Scalable System for Pushing all TLS Revocations to All Browsers. In *Proc. of IEEE Symposium on Security and Privacy*, 2017.

[112] T. Lauinger, A. Chaabane, S. Arshad, W. Robertson, C. Wilson, and E. Kirda. Thou shalt not depend on me: Analysing the use of outdated javascript libraries on the web. In *Proc of NDSS*, 2017.

[113] M. Lécuyer, G. Ducoffe, F. Lan, A. Papancea, T. Petsios, R. Spahn, A. Chaintreau, and R. Geambasu. Xray: Enhancing the web's transparency with differential correlation. In *Proc. of USENIX Security Symposium*, 2014.

[114] M. Lecuyer, R. Spahn, Y. Spiliopolous, A. Chaintreau, R. Geambasu, and D. Hsu. Sunlight: Fine-grained targeting detection at scale with statistical confidence. In *Proc. of CCS*, 2015.

[115] P. G. Leon, B. Ur, Y. Wang, M. Sleeper, R. Balebako, R. Shay, L. Bauer, M. Christodorescu, and L. F. Cranor. What matters to users?: Factors that affect users' willingness to share information with online advertisers. In *Proc. of the Workshop on Usable Security*, 2013.

[116] P. G. Leon, B. Ur, Y. Wang, M. Sleeper, R. Balebako, R. Shay, L. Bauer, M. Christodorescu, and L. F. Cranor. What matters to users?: Factors that affect users' willingness to share information with online advertisers. In *Proc. of the Workshop on Usable Security*, 2013.

[117] A. Lerner, A. K. Simpson, T. Kohno, and F. Roesner. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *Proc. of USENIX Security Symposium*, Austin, TX, 2016.

[118] T.-C. Li, H. Hang, M. Faloutsos, and P. Efstathopoulos. Trackadvisor: Taking back browsing privacy from third-party trackers. In *Proc. of PAM*, 2015.

[119] B. Liu, A. Sheth, U. Weinsberg, J. Chandrashekar, and R. Govindan. Adreveal: Improving transparency into online targeted advertising. In *Proc. of HotNets*, 2013.

[120] M. Malheiros, C. Jennett, S. Patel, S. Brostoff, and M. A. Sasse. Too close for comfort: A study of the effectiveness and acceptability of rich-media personalized advertising. 2012.

[121] M. Malheiros, C. Jennett, S. Patel, S. Brostoff, and M. A. Sasse. Too close for comfort: A study of the effectiveness and acceptability of rich-media personalized advertising. 2012.

[122] M. Malloy, M. McNamara, A. Cahn, and P. Barford. Ad blockers: Global prevalence and impact. In *Proc. of IMC*, 2016.

[123] J. R. Mayer and J. C. Mitchell. Third-party web tracking: Policy and technology. In *Proc. of IEEE Symposium on Security and Privacy*, 2012.

[124] A. M. McDonald and L. F. Cranor. Americans' attitudes about internet behavioral advertising practices. In *Proc. of WPES*, 2010.

[125] A. M. McDonald and L. F. Cranor. A survey of the use of adobe flash local shared objects to respawn http cookies. *ISJLP*, 7(639), 2011.

[126] R. McLean. A hacker gained access to 100 million capital one credit card applications and accounts, July 2019. https://www.cnn.com/2019/07/29/business/capital-one-data-breach/index.html.

[127] G. Merzdovnik, M. Huber, D. Buhov, N. Nikiforakis, S. Neuner, M. Schmiedecker, and E. R. Weippl. Block me if you can: A large-scale study of tracker-blocking tools. In *IEEE European Symposium on Security and Privacy (Euro S&P)*, 2017.

[128] WhiteOps - The Methbot Operation. WhiteOps, Dec. 2016. https://www.whiteops.com/methbot.

[129] A. Metwally, D. Agrawal, and A. E. Abbadi. Duplicate detection in click streams. In *Proc. of WWW*, 2005.

[130] A. Metwally, D. Agrawal, and A. E. Abbadi. Detectives: detecting coalition hit inflation attacks in advertising networks streams. In *Proc. of WWW*, 2007.

[131] B. Miller, P. Pearce, C. Grier, C. Kreibich, and V. Paxson. What's clicking what? techniques and innovations of today's clickbots. 2011.

[132] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proc. of IMC*, 2007.

[133] K. Mowery, D. Bogenreif, S. Yilek, and H. Shacham. Fingerprinting information in JavaScript implementations. In *Proc. of W2SP*, 2011.

[134] K. Mowery and H. Shacham. Pixel perfect: Fingerprinting canvas in html5. In *Proc. of W2SP*, 2012.

[135] Mozilla. Same-origin policy., May 2008. https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy.

[136] MTurk. Amazon Mechanical Turk. Amazon. https://www.mturk.com/, Accessed on 07/02/19.

[137] M. H. Mughees, Z. Qian, and Z. Shafiq. Detecting anti ad-blockers in the wild. *PoPETs*, 2017(3):130, 2017.

[138] M. Mulazzani, P. Reschl, M. Huber, M. Leithner, S. Schrittwieser, and E. Weippl. Fast and reliable browser identification with JavaScript engine fingerprinting. In *Proc. of W2SP*, 2013.

[139] H. Nazerzadeh, A. Saberi, and R. Vohra. Dynamic cost-per-action mechanisms and applications to online advertising. In *Proc. of WWW*, 2008.

[140] N. Nikiforakis, W. Joosen, and B. Livshits. Privaricator: Deceiving fingerprinters with little white lies. In *Proc. of WWW*, 2015.

[141] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Proc. of IEEE Symposium on Security and Privacy*, 2013.

[142] R. Nithyanand, S. Khattak, M. Javed, N. Vallina-Rodriguez, M. Falahrastegar, J. E. Powles, E. D. Cristofaro, H. Haddadi, and S. J. Murdoch. Adblocking and counter blocking: A slice of the arms race. In *Proc. of FOCI*, 2016.

[143] K. O'Donnell and H. Cramer. People's perceptions of personalized ads. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, 2015.

[144] F. of Privacy Forum. This year's must-read privacy papers: The future of privacy forum announces recipients of annual privacy papers for policymakers award, December 2018. https://fpf.org/2018/12/17/this-years-must-read-privacy-papers-the-future-of-privacy-forum-announce

[145] L. Olejnik. Enhancing user transparency in online ads ecosystem with site self-disclosures, Dec. 2018. https://lukaszolejnik.com/adstxt-transparency.pdf.

[146] L. Olejnik, C. Castelluccia, and A. Janc. Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns. In *Proc. of HotPETs*, 2012.

[147] L. Olejnik, T. Minh-Dung, and C. Castelluccia. Selling off privacy at auction. In *Proc of NDSS*, 2014.

[148] Openx announces new ads.txt policy banning all unauthorized resellers. Business Wire, Jan. 2018. https://www.businesswire.com/news/home/20180131005710/en/OpenX-Report-Finds-Ads.txt-Adoption-Accelerating-Majority.

[149] P. Papadopoulos, N. Kourtellis, P. Rodriguez, and N. Laoutaris. If you are not paying for it, you are the product: How much do advertisers pay for your personal data? In *Proc. of IMC*, 2017.

[150] F. Papaodyssefs, C. Iordanou, J. Blackburn, N. Laoutaris, and K. Papagiannaki. Web identity translator: Behavioral advertising and identity privacy with wit. In *Proc. of HotNets*, 2015.

[151] P. Pearce, V. Dave, C. Grier, K. Levchenko, S. Guha, D. McCoy, V. Paxson, S. Savage, and G. M. Voelker. Characterizing large-scale click fraud in zeroaccess. In *Proc. of CCS*, 2014.

[152] T. Peterson. Facebook's liverail exits the ad server business, January 2016. http://adage.com/article/digital/facebook-s-liverail-exits-ad-server-business/302017/.

[153] T. Peterson. Ads.txt has gained adoption, but 19 percent of advertisers still havenâĂŹt heard of it. Digiday, Apr. 2018. https://digiday.com/media/state-ads-txt-5-charts/.

[154] Ads.txt adoption: IABâĂŹs program grows 5.4x in 2018. Pixalate, Sept. 2018. https://blog.pixalate.com/ads-txt-adoption-trends.

[155] Ads.txt reduces ad fraud by 10%, but double-digit ad fraud rates persist. Pixalate, Sept. 2018. https://blog.pixalate.com/does-ads-txt-reduce-ad-fraud.

[156] A. C. Plane, E. M. Redmiles, M. L. Mazurek, and M. C. Tschantz. Exploring user perceptions of discrimination in online targeted advertising. In *26th USENIX Security Symposium (USENIX Security 17)*, 2017.

[157] A. Plus. What are acceptable ads without third-party tracking? https://adblockplus.org/acceptable-ads#privacy-friendly-acceptable-ads, Accessed on 07/12/19.

[158] V. L. Pochat, T. V. Goethem, S. Tajalizadehkhoob, M. KorczyÅĎski, and W. Joosen. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proc of NDSS*, 2019.

[159] E. Pujol, O. Hohlfeld, and A. Feldmann. Annoyed users: Ads and ad-block usage in the wild. In *Proc. of IMC*, 2015.

[160] US Online and Traditional Media Advertising Outlook, 2018-2022. Marketing Charts, June 2018. https://www.marketingcharts.com/featured-104785.

[161] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76, Sep 2007.

[162] A. Razaghpanah, R. Nithyanand, N. Vallina-Rodriguez, S. Sundaresan, M. Allman, C. Kreibich, and P. Gill. Apps, trackers, privacy and regulators: A global study of the mobile tracking ecosystem. In *Proc of NDSS*, 2018.

[163] R. Reagan. Why rtb will enable more advertisers to embrace mobile video, May 2017. https://www.businesswire.com/news/home/20161227005140/en/Growth-Video-RTB-Drive-Programmatic-Advertising-Display.

[164] M. Reavy. Webrtc privacy. mozillamediagoddess.org, Sept. 2015. https://mozillamediagoddess.org/2015/09/10/webrtc-privacy/.

[165] J. Ren, A. Rao, M. Lindorfer, A. Legout, and D. Choffnes. Recon: Revealing and controlling pii leaks in mobile network traffic. In *Proc. of MobiSys*, 2016.

[166] Technobuffalo.com. EasyList Forum, July 2016. https://forums.lanik.us/viewtopic.php?p=110902.

[167] N. Richter. Helping the industry prevent the sale of counterfeit inventory with ads.txt. IAB Tech Lab, May 2017. https://iabtechlab.com/blog/helping-industry-prevent-sale-of-counterfeit-inventory-with-ads-txt/.

[168] F. Roesner, T. Kohno, and D. Wetherall. Detecting and defending against third-party tracking on the web. In *Proc. of NSDI*, 2012.

[169] Buyers must stand up for ads.txt. rubiconProject, July 2018. https://rubiconproject.com/insights/technology/buyers-must-stand-up-for-ads-txt/.

[170] W. Rweyemamu, T. Lauinger, C. Wilson, W. Robertson, and E. Kirda. Clustering and the Weekend Effect: Recommendations for the Use of Top Domain Lists in Security Research. 2019.

[171] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez. A long way to the top: Significance, structure, and stability of internet top lists. In *Proc. of IMC*, 2018.

[172] S. Scott. The $8.2 Billion Adtech Fraud Problem That Everyone Is Ignoring. TechCrunch, Jan. 2016. https://techcrunch.com/2016/01/06/the-8-2-billion-adtech-fraud-problem-that-everyone-is-ignoring/.

[173] J. Shearer. Trojan.Zeroaccess. Symantec), July 2011. https://www.symantec.com/security-center/writeup/2011-071314-0410-99.

[174] M. Shields. Facebook buys online video tech firm liverail, looks for bigger role in digital ads, July 2014. https://blogs.wsj.com/cmo/2014/07/02/facebook-buys-online-video-tech-firm-liverail-looks-for-bigger-role-in-d

[175] G. P. Slefo. Ad Fraud Will Cost $7.2 Billion in 2016, ANA Says, Up Nearly $1 Billion. AdAge, Jan. 2016. https://adage.com/article/digital/ana-report-7-2-billion-lost-ad-fraud-2015/302201.

[176] P. Snyder, L. Ansari, C. Taylor, and C. Kanich. Browser feature usage on the modern web. In *Proc. of IMC*, 2016.

[177] G. Soeller, K. Karahalios, C. Sandvig, and C. Wilson. Mapwatch: Detecting and monitoring international border personalization on online maps. In *Proc. of WWW*, 2016.

[178] B. Software. Brave Rewards | Earn more for content you publish to the web. Brave. https://creators.brave.com/, Accessed on 06/29/19.

[179] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. J. Hoofnagle. Flash cookies and privacy. In *AAAI Spring Symposium: Intelligent Information Privacy Management*, 2010.

[180] L. Spector. Online privacy tips: 3 ways to control your digital footprint. PC World, January 2016. http://www.pcworld.com/article/3020163/internet/online-privacy-tips-3-ways-to-control-your-digital-footprint.html.

[181] K. Springborn and P. Barford. Impression fraud in on-line advertising via pay-per-view networks. In *Proc. of USENIX Security Symposium*, 2013.

[182] O. Starov and N. Nikiforakis. Extended tracking powers: Measuring the privacy diffusion enabled by browser extensions. In *Proc. of WWW*, 2017.

[183] B. Stone-Gross, R. Stevens, A. Zarras, R. Kemmerer, C. Kruegel, and G. Vigna. Understanding fraudulent activities in online ad exchanges. In *Proc. of IMC*, 2011.

[184] S. Tingleff. The Three Deadly Sins of ads.txt and How Publishers Can Avoid Them . IAB Tech Lab, May 2019. https://iabtechlab.com/blog/the-three-deadly-sins-of-ads-txt-and-how-publishers-can-avoid-them/.

[185] M. Trevisan, S. Traverso, E. Bassi, and M. Mellia. 4 years of EU cookie law: Results and lessons learned. *PoPETs*, 2019(2):126–145, 2019.

[186] M. C. Tschantz, S. Egelman, J. Choi, N. Weaver, and G. Friedland. The accuracy of the demographic inferences shown on google's ad settings. In *Proc. of WPES*, 2018.

[187] J. Turow, M. Hennessy, and N. Draper. The tradeoff fallacy: How marketers are misrepresenting american consumers and opening them up to exploitation. Report from the Annenberg School for Communication, June 2015. https://www.asc.upenn.edu/sites/default/files/TradeoffFallacy_1.pdf.

[188] B. Ur, P. G. Leon, L. F. Cranor, R. Shay, and Y. Wang. Smart, useful, scary, creepy: Perceptions of online behavioral advertising. In *Proc. of the Workshop on Usable Security*, 2012.

[189] N. Vallina-Rodriguez, J. Shah, A. Finamore, Y. Grunenberger, K. Papagiannaki, H. Haddadi, and J. Crowcroft. Breaking for commercials: Characterizing mobile advertising. In *Proc. of IMC*, 2012.

[190] R. J. Walls, E. D. Kilmer, N. Lageman, and P. D. McDaniel. Measuring the impact and perception of acceptable advertisements. In *Proc. of IMC*, 2015.

[191] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. Metzger, H. Zheng, and B. Y. Zhao. Social turing tests: Crowdsourcing sybil detection. In *Proc. of NDSS*, 2013.

[192] Whotracks.me - bringing transparency to online tracking. Cliqz GmbH. https://whotracks.me/.

[193] J. Wilander. Intelligent tracking prevention 2.1, February 2019. https://webkit.org/blog/8613/intelligent-tracking-prevention-2-1/.

[194] C. E. Wills and C. Tatar. Understanding what they do with what they know. In *Proc. of WPES*, 2012.

[195] C. Wilson, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. Beyond Social Graphs: User Interactions in Online Social Networks and their Implications. *ACM Transactions on the Web (TWEB)*, 6(4):5:1–5:31, November 2012.

[196] B. Wire. Growth in video rtb to drive the programmatic advertising display market until 2021, says technavio, Dec. 2016. https://www.businesswire.com/news/home/20161227005140/en/Growth-Video-RTB-Drive-Programmatic-Advertising-Display.

[197] S. Wolpin. International privacy day: Protect your digital footprint. The Huffington Post, January 2015. http://www.huffingtonpost.com/stewart-wolpin/international-privacy-day_b_6551012.html.

[198] A. Zarras, A. Kapravelos, G. Stringhini, T. Holz, C. Kruegel, and G. Vigna. The dark alleys of madison avenue: Understanding malicious advertisements. In *Proc. of IMC*, 2014.

[199] L. Zhang and Y. Guan. Detecting click fraud in pay-per-click streams of online advertising networks. In *Proc. of ICDCS*, 2008.

[200] S. Zhu, U. Iqbal, Z. Wang, Z. Qian, Z. Shafiq, and W. Chen. Shadowblock: A lightweight and stealthy adblocking browser. In *The World Wide Web Conference*, Proc. of WWW, 2019.

[201] S. Zimmeck, J. S. Li, H. Kim, S. M. Bellovin, and T. Jebara. A privacy analysis of cross-device tracking. In *Proc. of USENIX Security Symposium*, Vancouver, BC, 2017.

Chapter 8

Appendices

8.1 Clustered Domains

I clustered the following domains together when classifying publisher-side chains in § 4.3.1.2

Google: google-analytics, googleapis, google, doubleclick, gstatic, googlesyndication, googleusercontent, googleadservices, googletagmanager, googletagservices, googlecommerce, youtube, ytimg, youtube-mp3, googlevideo, 2mdn

OpenX: openxenterprise, openx, servedbyopenx

Affinity: affinitymatrix, affinity

Ebay: ebay, ebaystatic

Yahoo: yahoo, yimg

Mythings: mythingsmedia, mythings

Amazon: cloudfront, amazonaws, amazon-adsystem, images-amazon

Tellapart: tellapart, tellaparts

8.2 Selected Ad Exchanges

I select the ad exchanges shown in Table 8.1 from the *Inclusion* graph by thresholding nodes with out-degree ≥ 50 and in/out degree ratio r in the range $0.7 \leq r \leq 1.7$. One notable ommission from this list is Facebook. The dataset used in this study was collected in December 2015. Facebook

Table 8.1: Selected ad Exchanges. Nodes with out-degree ≥ 50 and in/out degree ratio r in the range $0.7 \leq r \leq 1.7$.

Node	**Out Degree**	**In/Out Ratio**
doubleclick	398	1.67
googleadservices	380	1.00
googlesyndication	318	1.28
adnxs	293	0.98
googletagmanager	253	0.98
2mdn	223	0.97
adsafeprotected	202	1.30
rubiconproject	191	1.14
mathtag	182	1.09
openx	170	0.79
pubmatic	157	0.96
casalemedia	136	1.10
krxd	134	1.08
adtechus	130	0.96
yahoo	124	1.31
chartbeat	124	0.96
contextweb	117	0.88
crwdcntrl	105	1.36
rlcdn	98	1.50
turn	86	1.48
amazon-adsystem	84	1.43
bzgint	72	0.86
monetate	72	0.76
rhythmxchange	71	1.13
rfihub	70	1.46
gigya	69	0.78
revsci	67	1.00
media	57	1.07
adtech	57	0.93
simplereach	57	0.84
tribalfusion	55	0.75
disqus	55	0.95
w55c	55	1.55
afy11	54	1.33
adform	52	1.62
teads	51	1.61

planned the shut down of its public ad exchange around that time [152], which it acquired from LiveRail in 2014 [174].