The Cost of Digital Advertisement: Comparing User and Advertiser Views

Panagiotis Papadopoulos
FORTH-ICS, Greece
panpap@ics.forth.gr

Nicolas Kourtellis
Telefonica Research, Spain
nicolas.kourtellis@telefonica.com

Evangelos P. Markatos
FORTH-ICS, Greece
markatos@ics.forth.gr

ABSTRACT

Digital advertisements are delivered in the form of static images, animations or videos, with the goal to promote a product, a service or an idea to desktop or mobile users. Thus, the advertiser pays a monetary cost to buy ad-space in a content provider's medium (e.g., website) to place their advertisement in the consumer's display. However, is it only the advertiser who pays for the ad delivery?

Unlike traditional advertisements in mediums such as newspapers, TV or radio, in the digital world, the end-users are also paying a cost for the advertisement delivery. Whilst the cost on the advertiser's side is clearly monetary, on the end-user, it includes both quantifiable costs, such as network requests and transferred bytes, and qualitative costs such as privacy loss to the ad ecosystem.

In this study, we aim to increase user awareness regarding the hidden costs of digital advertisement in mobile devices, and compare the user and advertiser views. Specifically, we built OpenDAMP, a transparency tool that passively analyzes users' web traffic and estimates the costs in both sides. We use a year-long dataset of 1270 real mobile users and by juxtaposing the costs of both sides, we identify a clear imbalance: the advertisers pay several times less to deliver ads, than the cost paid by the users to download them. In addition, the majority of users experience a significant privacy loss, through the personalized ad delivery mechanics.

CCS CONCEPTS

• **Information systems** → **Online advertising**; Web log analysis; • **Security and privacy** → *Economics of security and privacy*;

KEYWORDS

Cost of mobile advertising, Personalized advertising, User privacy

ACM Reference Format:
Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. 2018. The Cost of Digital Advertisement: Comparing User and Advertiser Views. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3178876.3186060

1 INTRODUCTION

The digital advertising business grew to $194.6 billion in 2016 [51] of which $108 billion were due to mobile advertising. In addition, it is digital advertising that fuels the internet as we know it. The vast majority of the content providers offer their websites or their sophisticated services free-of-charge (e.g. Google Docs, Facebook, Twitter, Gmail) in exchange for allowing third parties to access and display advertisements to their users.

Online advertising progressively moves towards more personalized ad delivery through a programmatic ad-buying model. In this model, advertisers buy available ad-slots in the user's display in an automated fashion based on how well the advertised product matches the profile of the user. As a consequence, when a user visits a website, each of the available ad-slots is auctioned, and advertisers decide if they will bid and how much, based on the information (interests, income, gender etc.) they have about the current user.

Following the above process, a careful reader identifies 3 key role players: (i) the website provider who earns money from advertisers through the auctions, (ii) the advertisers that pay to promote, and eventually sell their products by delivering effective advertisements to the proper eyeballs, and (iii) the user that receives from the website the content of his interest, for free. Seemingly, everyone benefits from this model. *But are the users indeed receiving the content they want free of charge?*

Contrary to the traditional advertising (i.e. in newspapers, TV, radio), in the digital world, it is not only the advertiser that pays the cost of advertisement delivery, but the user as well! Indeed, it is the user's data plan that is being charged to download the additional ad-related KBytes. To make matters worse, there are several other bytes the user downloads regarding analytics and user tracking, totally unassociated with the actual content of the visited website. Of course, the cost is not only monetary, since the privacy loss of the above operation has proven significant [33, 44].

In this study, we examine the hidden costs of mobile advertising for both the transmitter (advertiser), and the receiver (user) of the advertisments. In fact, we compare them for the same user profiles and investigate how fairly they are shared among the two sides. Our motivation is to enhance transparency regarding the overall costs of online advertising, and increase awareness of users regarding hidden costs they pay while using ad-supported online services.

Past works in the area already attempted to reveal the hidden costs of advertising in the mobile ecosystem. For example, Gui et al. [25] analyze free and paid version of apps to compare the advertising costs from the developers' side. They actively analyzed mobile apps to measure costs related to memory, power consumption and CPU usage. Similar to the study of Gao et al. [20], they compared these costs with the users' feedback from app reviews.

This work is the first to our knowledge that measures the hidden cost of advertising when mobile users browse the web. Contrary to the above inspiring approaches, our more user-centric study attempts to examine these costs, not from a developer perspective but from the side of the end-user. Towards this goal, we design a

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3186060

methodology and we implement it in OpenDAMP: a tool to es
the costs of advertising for both advertisers and users, by pa
analyzing a dataset of user HTTP traffic. We collected a d
consisting of mobile traffic from 1270 volunteering users tha
over an entire year, and use OpenDAMP to analyze it. Fina
compare the costs of both sides to assess how fair they are
across the two ends.

In summary, this work makes the following contributior

(1) We design a methodology to measure the costs a use when receiving ad-related traffic. These costs may ther directly quantifiable (e.g., requests, bytes, ener qualitative such as loss of privacy. In addition, and b on previous approach [45], our methodology estima costs advertisers pay to display each of the advertiser user receives through the contemporary programmat auctions [56].

(2) We implement our methodology in OpenDAMP (op ital Advertising Measurement Platform): a framew passive weblog analysis oriented to digital advertising. OpenDAMP can analyze user HTTP traffic and detect ID sharing incidents among third parties (known as Cookie Synchronizations). In addition, by incorporating information from external resources and blacklists, OpenDAMP can classify the traffic based on the content the domains deliver, and extract metadata and charge prices from RTB ad-auctions.

(3) To assess the effectiveness of our methodology, we collected a year-long dataset with mobile browsing traffic from 1270 volunteering users. Our analysis shows that the costs advertisers and users pay are largely unbalanced, In fact, users pay ~3 times more through their data plan to download ads, than what the advertisers pay to deliver them to these users. Furthermore, the majority of users sustains a significant loss of privacy to receive these personalized advertisements.

2 COST ANALYSIS WITH OPENDAMP

In this study, we measure the hidden costs of advertising for users, by passively monitoring their browsing traffic, while taking into account the advertisers' view. For our analysis, we set a server as proxy and recruited 1270 users located in the same country[1]. These volunteering users agreed to redirect their mobile web traffic through our proxy for 12 months. This way, we collected a year-long dataset of weblogs with a total of 250M HTTP requests (for security purposes we avoid breaking users' SSL connections).

2.1 Quantitative & Qualitative User Costs

Besides the quantitative costs a user may pay to receive advertisements, such as the additional network usage, there is also an important, qualitative cost for the user: the loss of privacy. It is well known that companies comprising the online advertising ecosystem collect several types of user data: location, behavior, preferences, interests, etc. Such data are used by these companies to deliver more personalized advertisements to online users.

[1] All Users, located in Spain, have signed a consent form allowing us to collect, analyze and publish the results extracted from their data.

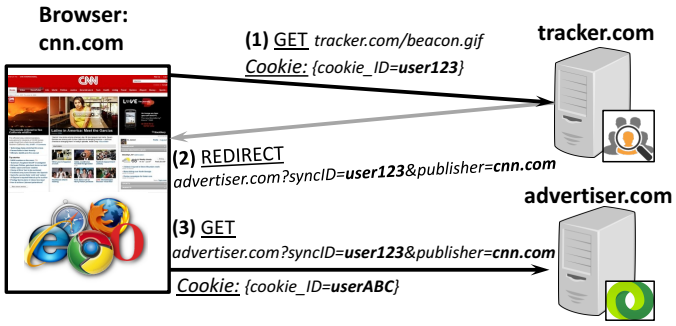

Figure 1: The CSync process in the wild. Two entities, through the user's browser, match the IDs they have set for the particular user.

Cookie Synchronization

In order for all this abundance of user data to be useful for the companies, there must be a *matching* process of all the userIDs that the third parties have assigned to the same user. By the notion userIDs, we define a string able to uniquely identify a user in the online world. In the web, the userID is an ID set on the user's side typically in the form of a cookie (i.e. cookie ID). Cookies, however, are domain-specific, which means those created by one third-party entity cannot be read by anyone else (see same origin policy [54]).

To remedy this, *Cookie Synchronization* (CSync) [1, 21, 34, 42] was invented, with which third parties are able to match the different userIDs they have set for the same user. Figure 1 presents a simple example to understand in practice how Cookie Synchronization works. Let's assume (i) a web site (say *cnn.com*), which includes some code from *tracker.com* and (ii) another third party site called *advertiser.com*, which is not included in the web page of *cnn.com* and thus, **does not (and cannot) know which users visit** *cnn.com*. Now, assume a user who, while browsing the web, got a cookie (cookieID=*user123*) by *tracker.com*, and another (cookieID=*userABC*) by *advertiser.com*, and now visits *cnn.com*. As soon as the code of *tracker.com* is called, a GET request is issued by the browser to *tracker.com* (step 1). Then, *tracker.com* responds back with a redirect request (step 2), instructing the user's browser to issue another GET request to *advertiser.com*, this time using a specifically crafted URL (step 3): *advertiser.com?syncID=**user123**&publisher=**cnn.com*** along with its cookie (cookieID=*userABC*).

When *advertiser.com* receives the above request along with the cookie ID *userABC*, it finds out that *userABC* visited *cnn.com*. To make matters worse, *advertiser.com* also learns that the user whom *tracker.com* knows as *user123* and the user *userABC* is basically one and the same user. Therefore, CSync enabled *advertiser.com* to collaborate with *tracker.com* in order to (i) find out which users visit *cnn.com* and (ii) synchronize (join) two different identities (cookie IDs) of the same user on the web.

Privacy Implications: There are significant privacy implications for online users raised by the above syncing process. By using CSync, in practice, *advertiser.com* learns (i) that whom it knew as *userABC* is also *user123* and (ii) that this user has just visited site *cnn.com*. This enables *advertiser.com* to track a user to a much larger number of sites than was initially thought. Indeed, by collaborating with several trackers, *advertiser.com* is able to track users across a wide variety of web sites, even if those web sites do not have any collaboration with *advertiser.com*. Last but not least, after the CSync,

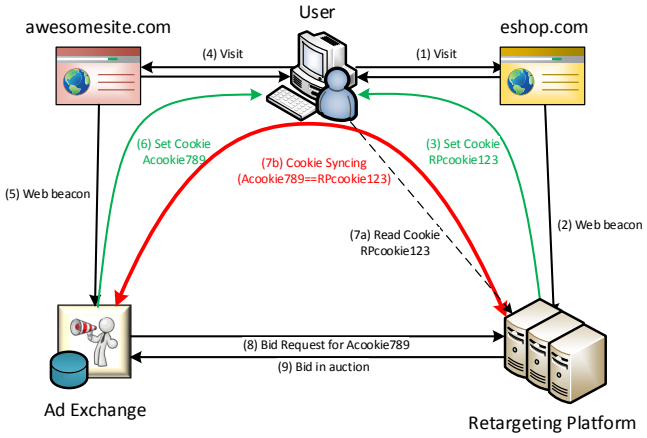

Figure 2: An example use of CSync in programmatic advertising. Advertisers can track and re-identify users while they surf the web.

tracker.com and *advertiser.com* can merge all data they have (and will have from now on) about this user. Nowadays, such cases of *server-to-server user data merges* take place at a massive scale [17], with the different web entities conducting mutual agreements for data exchanges or purchases, to enrich the quality and quantity of their user data warehouses [10, 35].

Thus, it is easy to anticipate, that the synchronized userIDs of Cookie Synchronization is of paramount importance for tracking entities in order to (a) re-identify users across the different websites they browse, but also (b) participate in user data auctions and marketplaces [2], thus increasing the wealth and detail of the information they know about each user. Thereby, in this study, we use CSync as a proxy for privacy loss. In fact, assuming 1 CSync leaks 1 userID, we use performed CSyncs as a metric to quantify and compare users' privacy and anonymity loss in mobile web.

Cookie Synchronization & Personalized Advertising
Besides user tracking, CSync, is also a core component of personalized advertising, which allows advertisers to re-identify (or retarget) users as they browse the web, and deliver them the proper ad. An example, as seen in Figure 2, is the following. Let's assume a publisher, e.g., a shoes-selling e-shop *E*, which collaborates with the Re-targeting Platform *RP* to improve the efficiency of its marketing strategy. In addition, let's also assume an Ad-Exchange *A*, with which *RP* is also collaborating. *RP* needs to be aware of the users visiting *E* at any time, as well as their movements: what other pages they visit, when and for how long. Therefore, *RP* asks *E* to tag each page of its website by embedding a *Web Beacon* [36, 40] pointing to the *RP* in each one of them: a 1-by-1 pixel image (also known as Pixel Tag or Web Bug). This way, the user will send this web beacon every time she browses the page, allowing *RP* to know her moves and also set a cookie (e.g. *UID_RP123*) on her side. Now, let's assume a user *U* who adds a pair of shoes in her shopping cart in *E*, but never makes it to the checkout. *E* would clearly want to re-target *U* and serve an ad, directing *U* back to *E* to try and finish the sale.

After a while, *U* surfs around the web, and lands on *awesomesite.com*, which is using *A* to monetize their ad inventory. Using a similar web beacon, *awesomesite.com* allows *A* to (i) learn about the visit of *U* and (ii) set a cookie *UID_A789*. Before *A* calls an auction for the available ad-slots of *awesomesite.com*, it trigger a Cookie Synchronization on *U*'s browser to share ID *UID_A789* with it's associated bidders (including *RP*). After this synchronization, *RP* can re-identify the user by matching the two aliases: *UID_A789* == *UID_RP123* and will bid accordingly to place a retargeting ad about the shoes of *E* that *U* intended to buy.

2.2 The OpenDAMP framework

To analyze our traffic, we built *OpenDAMP* (open Digital Advertising Measurement Platform): a framework for weblog analysis oriented to digital advertising. OpenDAMP parses HTTP traffic and classifies it based on the content delivered by the domains. In addition, using metadata from public crowd-based resources[2], it can further categorize advertisers based on the products they provide (DMPs, ad platforms, DSPs, SSPs, etc.). Finally, leveraging the User-Agent field of the HTTP requests, OpenDAMP can identify the operating system of the device (iPhone, WindowsPhone, Android) based on the set hardware characteristics.

Traffic classification: As we noted above, using OpenDAMP, we are able to classify the traffic into 5 categories (i) *Advertising*, (ii) *Analytics*, (iii) *Social*, which includes social widgets and plugins and (iv) *3rd party Content*, which includes content originated from 3rd party providers (for example content from CDNs, embedded Instagram photos, Captchas, blog comment hosting services like Disqus and many more) and (v) *Other*, which includes the rest of the content that is the useful content the user is actually interested in. To do such classification, OpenDAMP uses a popular browser adblock extension's blacklist [12]. This blacklist groups the different domains that belong to the same company (e.g. Google groups Doubleclick, AdMob and Adscape). It includes:

1) Advertising: 770 companies resulting in 1395 domains
2) Analytics: 150 companies resulting in 239 domains
3) Content: 111 companies resulting in 522 domains
4) Social: 17 companies resulting in 58 domains

CSync detection: To detect the Cookie Synchronization processes of our dataset, in OpenDAMP, first of all we extract all cookies set on the users' browsers. Then, inspired by previous works [42], we create a collection of heuristics aiming to extract all IDs shared among the entities which could possibly constitute a userID:

(a) We filter out the session cookies (cookies without expiration date) and we extract the userIDs that are able to uniquely identify the user.

(b) From the captured HTTP requests we keep only the ones with redirection status codes (i.e. 301, 302, 303).

(c) We identify ID-looking strings carried (i) as parameters in every request's URL, or (ii) in the referrer URL. As ID-looking strings, we define strings with specific length and number of alphas and digits (false positives do not matter at this point), that are unique for each user.

(d) Each of such ID-looking strings is stored upon detection in a hashtable along with the URL's domain (receiver of the ID).

[2]Business Software and Services reviews: g2crowd.com

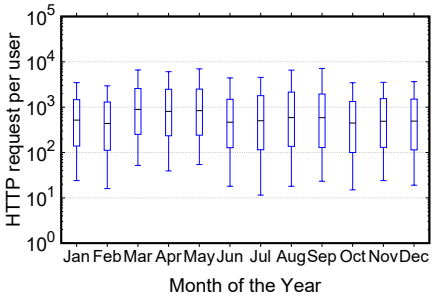

Figure 3: HTTP requests produced per user, across the year. Users create a relatively stable HTTP traffic, typically increased during holiday periods.

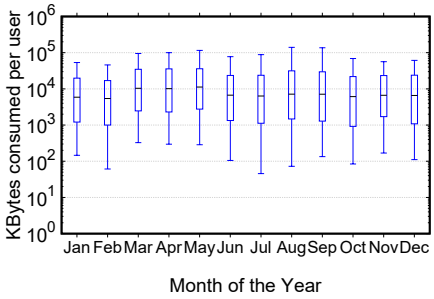

Figure 4: Volume of total consumed KBytes per user, across the year. Users consume an average of 5.9 GBytes per month.

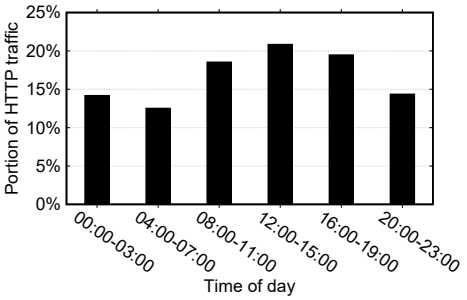

Figure 5: Portion of HTTP requests produced across the day. As expected, users produce web traffic mostly from morning till early afternoon.

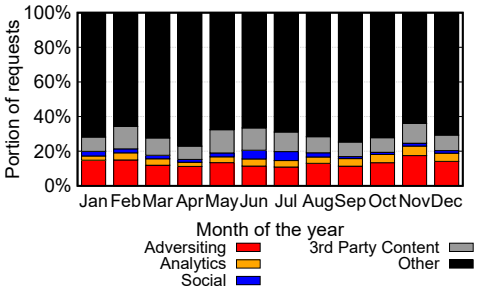

Figure 6: Portion of HTTP requests per content category the average user fetches through the year. On average, 77% of the HTTP requests is associated with the content the user is actually interested in.

(e) In case we have already seen the same ID in the past, we consider the two requests as a shared ID only if they are about different domains.

(f) To ensure that we capture, and exclude, cases of different domains owned by the same provider (e.g., doubleclick and googlesyndicate), we use several sources like DNS whois, blacklists etc. By filtering out domains of the same provider, our approach can discriminate between intentional ID leaking and unequivocally legitimate cases of internal ID sharing, thus, avoiding false positives.

(g) Finally, in order to verify if the detected shared ID is a userID able to uniquely identify a user, we search this ID in the list of the userIDs that we extracted in step (a). If there is a match, then we consider this request as a CSync.

3 THE VIEW OF THE USER

In this section, we analyze the costs that users sustain to receive advertisements while browsing the web. In our dataset, we separate the web traffic of each user and we compose user timelines that describe the traffic characteristics of each one of them. The timelines include HTTP requests received, Bytes transferred, files received, impressions received etc.

All the above constitute quantifiable properties that we can measure to extract the final cost a user paid. However, while browsing the web, users also leak information that is useful for the advertising ecosystem and this is another cost of advertising. In this section, we also attempt to quantify this cost besides its qualitative properties.

3.1 Network resources consumption

How many HTTP requests are due to ads?

First, we conduct a brief analysis to explore the contents of the collected dataset regarding the network traffic of the users. In Figure 3 and Figure 4, we see respectively the distribution of the overall HTTP requests produced and the KBytes consumed per user through the year in our dataset (percentiles: 10th, 25th, 50th, 75th, 90th). As we see, the median user has a relatively steady production of network traffic, thus consuming per month around 5891 KBytes, on average. In addition, we see an expected monthly behavior, where there is an increase of the produced web traffic during months that include long holidays (spring break, summer holidays etc.). A diurnal behavior can be also seen when measuring the time of day the traffic was produced. As shown in Figure 5, users produce web traffic in their mobile devices mostly from morning till early afternoon, and this repeats throughout the week.

In Figure 6, we use OpenDAMP to classify the HTTP requests the average user fetches, based on the content served by their domain, across the whole year. Considering that 3rd party content is an essential (external) component of a website and its absence could break the provided functionality and degrade the experience of the user, we consider it as part of the actual content of the website. On the other hand, the Analytics category includes services which aim to monitor performance and behaviorally track the audience of a website. Thus, we see that the percentage of requests bringing to the user's browser the actual content they are interested in is steadily around 77% across the whole year, and the percentage of ad- and analytics- related percentage is as high as 19%, on average.

Next, in Figure 7, we investigate what are the different resources a user retrieves for these two content categories through the year. In this plot, we present the distribution of the users (percentiles: 10th, 25th, 50th, 75th, 90th). For the median user, most of the advertising HTTP requests are animated and static images and scripts, besides the expected volume of HTML. Also, in analytics, the largest amount of requests are monitoring scripts.

How much of the downloaded volume is related to ads?

The cost for all of the above (additional) resources the user downloads is translated to consumed Bytes. This is the most important

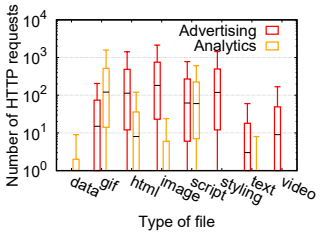

Figure 7: HTTP requests received per user, per different resource type.

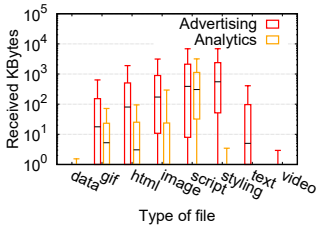

Figure 8: Bytes received per user, per different resource type.

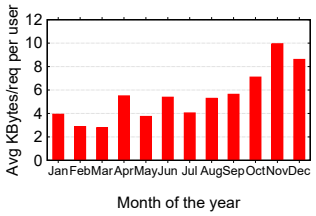

Figure 9: KBytes per ad-related HTTP request per user, across the year.

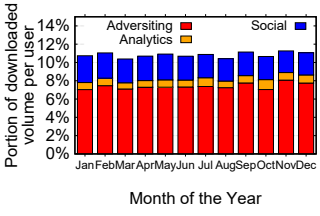

Figure 10: Ad-related KBytes downloaded per user, through the year.

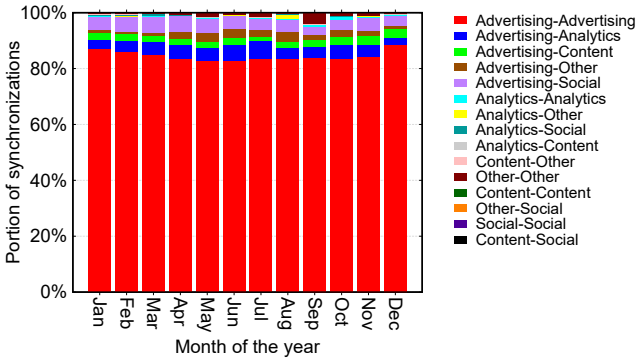

Figure 11: Portion of CSyncs per content category pair, through the year.

metric that not only monetarily affects the user's data plan, but affects also the device's battery by keeping its CPU and network card on, in order to marshal the received content. From Figure 8, it is evident that the volume of bytes for the downloaded static advertising images and scripts reaches around 700 KBytes and 850 KBytes, respectively; the 90th percentile peaks at almost 10 MByte for each one. It is easy to observe in these two Figures (Figure 7 and Figure 8) the large amount and size of the scripts that both Advertising and Analytics related domains instruct the user's browser to run. Note that these scripts, and the additional CPU cycles they require, are unrelated with the actual content the user is interested in, and therefore constitute a clear additional overhead for the user.

If we have a deeper look in the HTTP requests and the volume of bytes they deliver, in Figure 9 we observe an increasing trend across the year, with the HTTP requests for ads requiring to transfer double the volume, on average (from 4KB to 8KB). Taken in conjunction with Figure 6 which shows a steady portion of ad-related requests, delivering larger payloads in the same number of requests, although it may require more memory from the device, it gives the opportunity for the device to minimize the required latency to marshal/unmarshal each ad-related requests. However, we also suspect that advertisers take advantage of better mobile network speeds and device resources, as they become available through time. Consequently, they force each mobile device to download an ever-increasing amount of data displayed in the publishers' pages, at the users' expense.

Finally, we measure the portion of the total downloaded volume per user that is associated to Advertising and Analytics. In Figure 10, we see that a user steadily downloads an average 8.2% of bytes (extra to the actual content they browse) across the year, which belongs solely to Advertising (7.3%) and Analytics (0.8%) related content. We see a small increase in the ad-related volume with previous studies (5 years ago) [52] measuring the same volume at 5.6%. If we also add the Social-related traffic, the total percentage of additional content the user has to download reaches as high as 11%, on average.

Using the results from [24, 41, 55], we also provide an estimation of the power the ad-related traffic consumes on the user side. Given the results in Figure 10, the network component of a mobile device alone consumes 7.98% more, due to the additional ad-related transmitted bytes, and 0.86% more, due to analytics-related bytes. This means that a mobile device, whose battery can sustain 10 hours of ad-free browsing, will last 9.2 hours due to the additional ad-related network volume received. In fact, and according to previous studies [25], if we also consider the energy consumption of the display, this cost may surpass 15%.

Unlimited data plans

Passively measuring the cost on the users' data plans, of course, comes with some limitations. First of all, there may be user devices connected to the Internet through WiFi. In addition, some ISPs recently offered *unlimited data plans*, providing a large volume of data (usually around 20 GB/month [26]) to their clients. Despite the current issues of such products (i.e., throttling [50], high prices (70-90$/month) [26], expensive Internet roaming), it is likely that in the future they will become cheap enough to become popular. Therefore, the respective monetary cost for users with unlimited data plans will become practically negligible. However, even in such cases, personalized advertisements do consume device resources (battery, network traffic, CPU, etc.), and still incur a high cost on user privacy and anonymity loss.

3.2 User privacy loss

What is the user's exposure to Cookie Synchronization?

By using OpenDAMP, we detect CSyncs in our dataset and we see that for users with regular activity on the web (> 10 HTTP requests per day), 97% of them were exposed to CSync at least once. Next, we separate and classify the pairs of entities that conduct CSync in our dataset through the year and in Figure 11 we show the portion of CSyncs performed by each type of pair. The majority (~85%) of the CSync takes place within the different advertising entities, but there are also cases where advertising entities synchronize their userIDs with Social or Analytics related entities.

Next, we investigate if the synchronizations the users are exposed to change over time. Hence, we extract CSyncs per user, normalized by the user's total number of requests. In Figure 12,

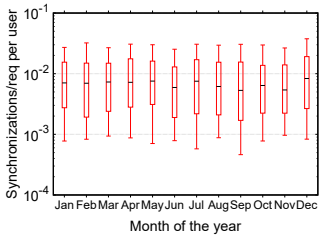

Figure 12: Synchronizations per HTTP request users receive through the year. The median user is exposed to a steady number of CSyncs.

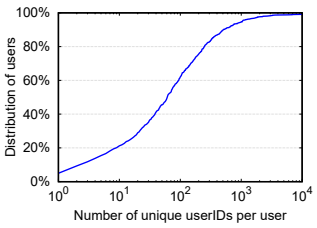

Figure 13: Unique synced userIDs per user. The 50th (75th) percentile user gets up to 63 (195) unique IDs synced, at least once.

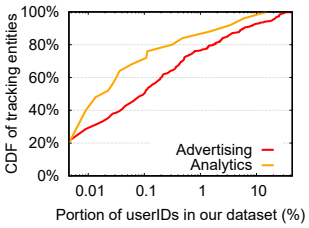

Figure 14: Portion of the overall userIDs in our dataset each tracking entity learned. Some entities have learned more than 10% of all userIDs.

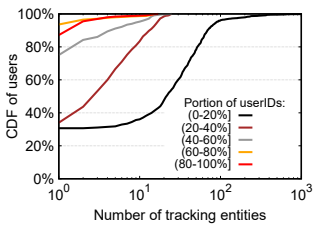

Figure 15: Number of entities having access to a portion of a user's IDs. The median user loses up to 20% of its anonymity to 22 tracking entities.

we plot these synchronizations across the year. The median user receives 1 synchronization per 140 HTTP requests, while the 90th percentile user is exposed to 1 synchronization per 50 requests! Considering the different userIDs that tracking entities may assign to a user, in Figure 13, we measure the number of unique userIDs that got synced per user. Evidently, a median user gets up to 63 different userIDs synced (at least once) through the year, and the 75th percentile user gets up to 195 of their userIDs synced.

How much do tracking entities know about a user?
Next, we measure the pervasiveness of the tracking entities. Specifically, in Figure 14, we measure the portion of the overall userIDs each (ad- and analytics- related) entity learned through CSync. Interestingly, ad and analytic entities follow similar distributions, and apparently, such entities tend to learn significant portions of userIDs. Therefore, although a median ad-related entity may learn around 0.03% of the overall userIDs, there is a portion of 5% of entities than learned more than 10%, and another 0.6% of entities that learned more than 25% of the overall userIDs in our dataset.

As we described earlier, CSync is a mechanism for trackers to increase the identifiability of a user in the web, by joining their assigned userIDs. In Figure 15, we plot the number of entities that gained access to the user's IDs. As we see, the median user loses up to 20% of her anonymity to 22 tracking entities and up to 40% to 3 tracking entities. Such an important leak enables a handful of entities to accurately re-identify the user on the web and construct a rich user profile by merging their collected data on the backend.

4 THE VIEW OF THE ADVERTISER

It is of no doubt that digital advertising moves towards a more personalized ad-delivery approach, where advertisements are matched to the interests of the individuals following a programmatic ad-buying model. The most popular one is the model of programmatic auctions of the Real-Time Bidding (RTB) [22], which has a five-year CAGR of 24% [4]. In RTB, ad-slots on the users' displays are being sold in auctions where the higher bidder delivers its impression.

More specifically, in RTB-based auctions, whenever a user visits a website with an available ad slot, an ad-request is sent to an Ad Exchange (ADX), which calls an auction and sends bid requests (along with user info) to ad-buyers (bidders). These bidders in RTB are usually Demand Side Platforms (DSPs), which are agencies that utilize sophisticated decision engines and aim to assist advertisers to decide at real time if they will bid at an auction and how much, based on the user info they receive and how close the advertised product is to the user's interests. The entire auction has a strict time constraint and usually takes 100 ms from the time that the user will visit the site till the winning impression is finally delivered.

In this paper, we leverage mobile RTB to assess the cost that advertisers pay, in order to deliver personalized ads to users. For this, we search for a specific step of the RTB where the ADX notifies, through the user's browser, the higher bidder about its win. Typically, this notification URL is parametrized with a keyword agreed between the two companies (ADX and DSP), and carries the RTB price to be paid by the winning DSP. The price can be cleartext or encrypted, as shown in two examples in Table 1.

Although the RTB protocol is well standardized by OpenRTB [27] since 2010, in Figure 16 we observe a large heterogeneity of keywords used to define the charge price. In fact, each ADX may use its very own parameter, making the RTB process less transparent, and more difficult for an external observer to detect and study the RTB parameters and values used.

We employ OpenDAMP and use pattern matching with publicly available lists of keywords from past studies [42, 45] and RTB documentations [14, 27, 28, 37, 43, 47, 48], and manage to extract a total of 44997 cleartext and encrypted charge prices across all users in our dataset. These impressions come from over 770 different advertisers and ad-networks. In addition, with OpenDAMP, we extracted the required features to estimate the value of the encrypted RTB prices. These features include user information that an ADX can provide to the bidders (user location, date and time of website visit, type of user device, user interests, etc.). Leveraging the technique in [45] and the extracted features, we computed an aggregated estimate for the advertiser cost per user, across the year, using both encrypted and cleartext prices.

In Figure 17, we present the RTB market share of each bidder in our dataset. As we can see, from the market share segmentation there is only a handful of big players winning the larger portion of auctions. Specifically, no more than 5 companies have won 67.7%

Winning Price Notification URLs
(A) cpp.imp.mpx.mopub.com/imp?ad_domain=amazon.es& ads_creative_id=ID&**bid_price=0.99**&bidder_id=ID&... &bidder_name=..&**charge_price=0.95**&country=..&...
(B) tags.mathtag.com/notify/js?exch=ruc&... &**price=B6A3F3C19F50C7FD**&... &3pck=http%3A%2F%2Fbeacon-eu2.rubiconproject.com%2F...

Table 1: Cleartext and encrypted RTB price notification examples.

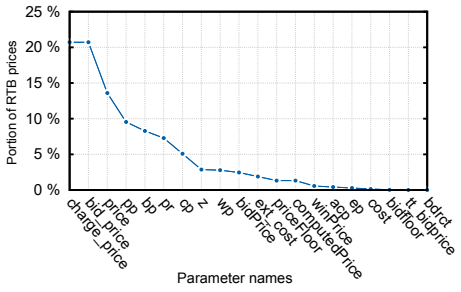

Figure 16: Although there is an OpenRTB standard [27], every company follows its very own protocol with different parameter naming, making RTB price filtering a challenging task.

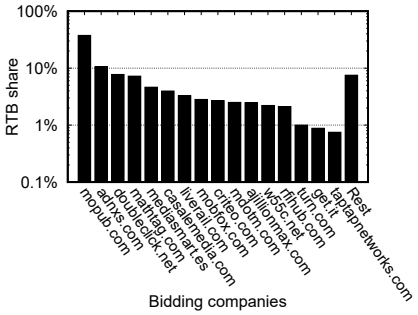

Figure 17: The RTB market share of the different bidders in our dataset. As we see, the market share is mainly divided to a dozen of companies with the top 5 wining 67.7% of the RTB auctions.

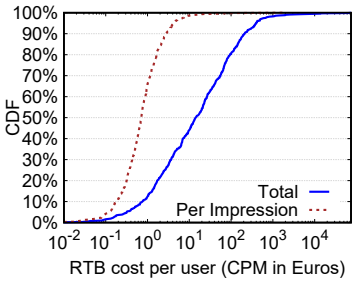

Figure 18: Cost per user for advertisers to display ads across the year. The average cost per impression for the median user is 0.9 CPM. The total cost paid by advertisers for the median user is ~22 CPM.

of the overall RTB auctions. In addition, we see only 14 of the total number of bidders in our sample, winning a portion of auctions greater or equal to 1%.

In Figure 18, we show the CDF of the total cost paid by advertisers to deliver and display ads to the mobile users of our dataset. These prices (in blue) represent what we have detected and computed as the total cost across the year for each user in our dataset, and expressed in CPM. As we can see, some users are orders of magnitude more costly to reach than the average user: advertisers paid for the 75th percentile user up to 100 CPM for the entire year, when they paid around 20 CPM for the median user.

In the same figure, we also plot the distribution of the costs per impression per user (in red). We see that an impression for the median user costs 0.9 CPM, but it is interesting to see that there are three classes of users: the users who are quite cheap to reach and are below average (<1CPM), the average users that can be reached with around 1 CPM, and the more expensive users (>1CPM) that advertisers paid up to 9 CPM per impression.

At this point, we must note that the above computed RTB charge prices regard only the value that a bidder paid for the specific ad-slot in a specific user's display. Commissions for possible intermediate agencies and platforms may appear, thus, increasing the actual cost that the advertised company may have paid.

5 CONSOLIDATING THE TWO VIEWS

Earlier, we showed how much advertisers paid to deliver ads to users, through various RTB ad-campaigns and companies. In this section, we use this RTB cost as a proxy for the monetary cost of the entire advertising process (e.g. user tracking, analytics and finally ad retrieval). We compare it with an estimated cost paid by the users to download these corresponding ads in their device. In particular, we use an estimation of the cost per byte that users paid in their data plans for the total bytes downloaded for these ads. We also look at the privacy cost of users via the CSync metric, and how that also compares with the advertisers' RTB cost.

5.1 Cost on data plan vs. Cost of RTB

For this comparison, we use currently available prices [3, 18], for various data plans in the country the users were located, while the dataset was collected. Using prices for 20 different data plans from 6 different ISPs and subsidiaries, we computed an average

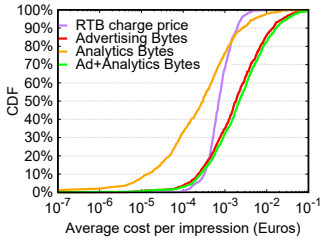

Figure 19: CDF of the average cost on the users' data plan, and cost paid by advertisers to deliver personalized ads to the same users.

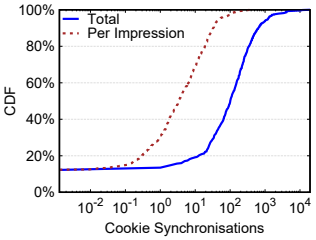

Figure 20: CDF of the average CSyncs per impression retrieved per user, across the year.

cost of Euros per Byte. Historically, the data plan prices have been dropping, thus, our estimation of the Byte cost can be considered a lower bound of the actual cost users paid during the data collection.

In Figure 19, we plot the CDF of average cost per impression paid by the two parties considered: (i) the end-users for Bytes consumed by their phones for downloading advertising and analytics requests, and (ii) the advertisers for ads they delivered to these devices through the RTB mechanism. These average scores reflect the traffic across the year. We make the following observations. Surprisingly, the cost on advertising bytes for the majority (about 80%) of users is higher than the RTB cost paid by the advertisers. Specifically, we see that the median user paid an average cost of 0.0022 Euros per ad for advertising and analytics bytes, whereas the median advertiser paid 0.00071 Euros per ad. This means that for each delivered ad impression, **users are charged 3 times more than advertisers** who benefit from the ad delivery!

Furthermore, we look at the average cost users pay for being delivered ads vs. the corresponding average cost advertisers paid for the exact same ads, for each user via a heatmap in Figure 21(a). We observe that the counts are skewed towards the upper left triangle for many of the users. In total, 67.4% of users paid more in bytes than what the advertisers paid for the same ads to be delivered. This means that the majority of mobile users pay more in data plan cost to download each impression (or even in total through the year), than the corresponding cost that advertisers pay to send the given ads displayed.

(a) Cost on data plan vs. Cost of RTB

(b) Cost of CSync vs. Cost of RTB

Figure 21: Heatmaps of (a) average cost per impression for Bytes consumed by users in advertising requests, (b) average Cookie Synchronizations per impression, both compared against the average cost paid by advertisers to deliver RTB ads to the same users (1-1 mapping), across the year.

5.2 Cost of Privacy vs. Cost of RTB

In section 3.2, we analyzed the cost of privacy for mobile users given the CSyncs performed by the advertising ecosystem. We measured how prevalent this practice is across users and through time. Here, we compare this privacy loss with the cost paid by advertisers in RTB ads delivered to users during the same time period.

In Figure 20 we show the CDF of the average CSyncs per impression (total CSyncs through the year in Figure 20 (line in blue)) that were performed through each user's device. We notice that the median user had about 3.4 synchronizations per impression, and 101 in total through the year. As explained earlier, this leads to loss of privacy to multiple third party companies.

We compare this cost on user privacy to the cost paid by advertisers with a heatmap in Figure 21(b). We notice that the main mass of the distribution of users cluster between 1 and 100 synchronizations per impression delivered (as also evident from Figure 20 (line in red)) and cost for the advertisers between 0.0005 and 0.001 Euros, per impression delivered. Also, in totals across the year, users have been exposed to 10-1000 synchronizations for all the ads they received, and these delivered ads cost between 0.005 and 0.05 Euros to the advertisers.

6 RELATED WORK

There are several studies aiming to measure different aspects and hidden costs of the advertising ecosystem. Gui et al. in [25] measure the cost of mobile advertisements to the mobile application developer by performing an empirical analysis of 21 apps. The authors consider several types of costs: (i) app performance, (ii) energy consumption, (iii) network usage, (iv) maintenance effort for ad-related code and (v) the user' feedback from app reviews. Their results show that apps with ads consume, on average: 48% more CPU time, 16% more energy, and 79% more network data. In addition, they found that the presence of ads in the apps affected the users' overall opinion leading to reduced ratings for the app.

Towards the same direction, Gao et al., propose IntelliAd [20], a tool to automatically measure the ad costs based on the different ad integration schemes. Similar to the above work, IntelliAd aims to provide developers with suggestions on how to better integrate ads into apps based on the costs the users are concerned. To identify the opinion of the users, the authors utilize several user reviews from 104 popular apps of Google Play. The types of the ad costs the users were concerned more include: number of ads, memory/CPU overhead, traffic usage, and battery consumption.

In [53] the authors quantify the network usage and system calls related to mobile ads, based on specific rules, aiming to quantify the difference between free and paid versions. In particular, they built a tool to profile apps at four different layers: (i) static, or app specification, (ii) user interaction, (iii) operating system, and (iv) network. They evaluate their approach by analyzing 27 free and paid Android apps. Their results show discrepancies between the app specification and app execution, as well as cases were free versions of apps were more costly than their paid counterparts due to their important increase in traffic. Finally, they observe that most network traffic is not encrypted and, even worse, apps communicate with many more sources than users might expect (as many as 13).

In [52], they analyze the characteristics of mobile ads by collecting a large volume of traffic of a major European ISP with over 3 million subscribers. Their results show that ad-related traffic is a significant portion of the overall traffic, and the associated market share is dominated by no more than 3 big ad-networks. In addition, they evaluate the energy consumption of three popular ad networks using a custom-built app with an ad slot at the bottom of the screen.

In [6], they analyze the browsing activity of a large sample of Internet users aiming to assess the impact of ad-blockers and regulatory policies which limit the use of third-party data for targeted advertising. Their results show that retailers attract only a small percentage (3%) of their customers through display ads. Although many publishers use ads as their main source of income, which makes them vulnerable to ad-blockers, browsing patterns suggest that ad revenue can generally be replaced by a small fraction of loyal visitors paying a modest subscription fee (e.g. $2 per month).

Apart from the academic studies, there is also an increased interest regarding the cost of the advertising ecosystem from the side of journalists and major news sites. For example, in [23] the editorial team conducted a small study measuring the estimated load time and data usage before and after blocking ad-related content on 50 popular news websites. Their results show that more than 50% of all data came from ads and other content.

Contrary to the above studies, our more user-centric approach provides a methodology to measure the hidden costs of advertising through passive monitoring of the users' traffic. We compare the cost users sustain, with the cost the advertisers pay for the ad delivery. Finally, we not only measure the monetary and network costs of digital advertising, but also the implications in privacy and anonymity of the users on the Internet via Cookie Synchronization.

7 DISCUSSION & CONCLUSION

Learnings: Unlike traditional advertising, in online mediums advertising imposes costs not only to the one who wants its message to be spread (the advertiser), but also to the one that receives it (the user). To make matters worse, the growth of personalized advertisement, where the advertisements are matched to the interests of the individuals, impose an additional cost for the users: the cost on their privacy and loss of anonymity.

In this study, we compare the costs on digital advertising for the advertiser and the user, in an attempt to identify how equal, or even comparable these costs are. Surprisingly, our results show that these costs are unbalanced, with the majority of users sustaining a significant loss of privacy, when the monetary cost they pay is, on average, 3 times more than what the advertisers are charged to deliver the given ads. Our findings can be summarized as follows:

- Ad- and analytics- related traffic is 19% of the total requests, and 8.2% of data plan volume of an average mobile user.
- Ad-related volume has been steadily increasing through the year, doubling from 4 KBytes to 8 KBytes per ad-request.
- Ad- and analytics- related traffic can potentially consume up to 9% of the phone's power, considering only the additional network overhead.
- 97% of regular mobile users are exposed to Cookie Synchronization at least once in a year.
- The 50th (75th) percentile user is exposed to one CSync every 140 (50) traffic requests, or every 3-4(1-2) website visits.
- The 50th (75th) percentile user gets up to 63 (195) of their unique user IDs synced in a year, at least once.
- Top 5% (0.6%) of ad-companies learn more than 10% (25%) of all user IDs, through the year.
- The median user loses up to 20% of their anonymity to 22 tracking entities, and up to 40% to 3 tracking entities.
- The top 5 ad-companies dominate 68% of RTB auctions.
- Mobile users are exposed to 10-1000 synchronizations for ads received through the year, which cost to the advertisers 0.005-0.05 Euros.
- The median advertiser paid 0.00071 Euro per delivered ad, but the median user paid 0.0022 Euro per ad in downloaded bytes.

Impact of Advertising Cost

Our results showed that in aggregate, and monetarily, over 2/3rds of users pay more through their data plan for downloading bytes related to ads and analytics, than the advertisers who sent the ads in the first place. In addition, given that: 1) the median user loses up to 20% of its anonymity to 22 tracking entities, 2) the top 5 ad-companies win the great majority of RTB auctions, and 3) these companies can sell the acquired data to 4th party companies in a non-transparent and backend fashion [35, 46], the loss of privacy experienced by an average user can be multiple times higher than that conservatively measured so far. Unfortunately, this pervasive user tracking effort to deliver more targeted impressions, fails to increase the effectiveness of the delivered ads. In fact, and according to [49], the average person is served over 1700 ads per month, but only half of them are ever viewed, and click rates for display ad campaigns reach 0.1% on average (i.e., one in a thousand impressions in a campaign is ever clicked). Furthermore, Budak et al. in [6] show that retailers attract only 3% of their customers through digital ads. Therefore, even though someone could argue that the user receives value from free access to the websites supported by advertisers, the amount of ineffective ads delivered to user devices is currently extreme, and costly for the end-user.

Considering all the above, the cost on the user's side with respect to 1) device resources spent for processing and displaying ads, 2) bytes downloaded and paid to the user's data plan, 3) loss of privacy experienced by the average user, all significantly outweigh both the efficiency of the received ads, and the cost paid by the ad ecosystem to deliver them to the user's device. Thus, it remains unclear whom the current advertising model benefits, apart from the ad-delivery and targeting companies.

Reducing or rebalancing the costs

Evidently, the annoyance, the inefficiency and the increased cost of advertisements have made users take measures to reduce the unbalanced costs they pay. The most popular of such actions is the use of mobile [13, 44] or desktop based [7, 12] ad-blockers. However, there are concerns [8, 29] that such all-out approaches are non-vital for the free Internet, as they significantly reduce the income of the ad-supported content providers, making them stop serving ad-blocking users [15, 38].

Approaches able to strike a vital middle-ground and rebalance the costs between advertisers and users, include Personal Information Management Systems (PIMS) [9, 31, 32, 39]. In PIMS, the user controls the privacy they expose to the online world, in return for a free service. A different approach is third-party ad-replacement systems [5] such as the Brave Browser [16], where the user gets compensated for each ad they receive. In addition, there is the CAMEO middleware [30], which aims to pre-fetch context-sensitive advertisement by predicting user context and pro-actively identifying relevant advertising content. This way, it can opportunistically use inexpensive wireless networks (e.g., WiFi) to predictively cache advertisement content on the mobile device.

The contribution of our work is to shed light upon the actual costs of ad-supported web. This way, we enhance the awareness of users regarding costs that they can easily measure (e.g., on their data plan), or cannot measure (e.g., privacy loss), in an attempt to help them choose between a seemingly free, ad-supported website and its paid ad-free counterpart [57].

Our future steps include active analysis of the user devices in order to measure additional hidden costs of advertising, that appear in power consumption, main memory, CPU. We will also study the impact advertising has on user experience by measuring the imposed latency due to the rendering time of digital ad impressions. In addition, active analysis on crafted user personas will allow us to determine the user data that get leaked together with the userIDs and if this is compliant with COPPA [19] rules and DAA's AdChoices [11] program.

Acknowledgments

The authors would like to acknowledge the contributions and help received by Dr. Matteo Varvello during the execution of this project. The research leading to these results has received funding from the European Union's Marie Sklodowska-Curie grant agreement No 690972 (project PROTASIS). The paper reflects only the authors' view and the Agency and the Commission are not responsible for any use that may be made of the information it contains.

REFERENCES

[1] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. 2014. The Web Never Forgets: Persistent Tracking Mechanisms in the Wild. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 674–689.

[2] Adthink S.A. 2018. BIG, The User Data Exchange. https://big.exchange/. (2018).

[3] AT&T. 2018. Create your Mobile Share Advantage plan. https://www.att.com/shop/wireless/data-plans.html#fbid=EhuxYcdIz02. (2018).

[4] BI Intelligence. 2015. The Programmatic-Advertising Report: Mobile, video, and real-time bidding drive growth in programmatic. http://www.businessinsider.com/buyers-and-sellers-have-overwhelmingly-adopted-programmatic-with-mobile-leading-growth-2015-3. (2015).

[5] Brave Software Inc. 2016. What is Brave Ad Replacement? https://www.brave.com/about-ad-replacement/. (2016).

[6] Ceren Budak, Sharad Goel, Justin Rao, and Georgios Zervas. 2016. Understanding emerging threats to online advertising. (2016).

[7] David Cancel, Felix Shnir, Alexei Miagkov, and Jose Maria Signanini. 2010. Ghostery Makes the Web Cleaner, Faster and Safer! https://www.ghostery.com/. (2010).

[8] Sean Captain. 2016. This Startup Wants To End Adblock Plus "Raping and Pillaging" Of Online Publishers. https://www.fastcompany.com/3055827/this-startup-wants-to-end-adblocks-raping-and-pillaging-of-online-publishers. (2016).

[9] Amir Chaudhry, Jon Crowcroft, Heidi Howard, Anil Madhavapeddy, Richard Mortier, Hamed Haddadi, and Derek McAuley. 2015. Personal data: thinking inside the box. In *Proceedings of The Fifth Decennial Aarhus Conference on Critical Alternatives*. Aarhus University Press, 29–32.

[10] Tom Chavez. 2010. Data: Deja Vu All Over Again? https://adexchanger.com/considering-digital/tom-chavez/. (2010).

[11] Digital Advertising Alliance. 2018. YourAdChoices Gives You Control. http://youradchoices.com/. (2018).

[12] Disconnect. 2018. Disconnect open-source browser plugin. https://disconnect.me/. (2018).

[13] Bruce Bujon Dominik Schurmann. 2012. AdAway open source ad blocker for Android. https://adaway.org/. (2012).

[14] DoubleClick. 2016. RTB Decrypt Price Confirmations. https://developers.google.com/ad-exchange/rtb/response-guide/decrypt-price. (2016).

[15] Editors of Wired Magazine. 2016. How WIRED Is Going to Handle Ad Blocking. https://www.wired.com/how-wired-is-going-to-handle-ad-blocking/. (2016).

[16] Brendan Eich, Brian R. Bondy, Marshall Rose, Yan Zhu, Garvan Keeley, Aubrey Keus, Sergey Zhukovsky, Brian Johnson, Brian Clifton, and Cezar Augusto. 2015. Brave free and open-source web browser. https://brave.com/. (2015).

[17] Steven Englehardt and Arvind Narayanan. 2016. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 1388–1401.

[18] FANDOM Lifestyle Community. 2017. Prepaid Data SIM Card Wiki - Spain. http://prepaid-data-sim-card.wikia.com/wiki/Spain. (2017).

[19] Federal Trade Commission. 2000. Children's Online Privacy Protection Act (COPPA). https://www.ftc.gov/enforcement/rules/rulemaking-regulatory-reform-proceedings/childrens-online-privacy-protection-rule. (2000).

[20] Cuiyun Gao, Yichuan Man, Hui Xu, Jieming Zhu, Yangfan Zhou, and Michael R. Lyu. 2017. IntelliAd: Assisting Mobile App Developers in Measuring Ad Costs Automatically. In *Proceedings of the 39th IEEE International Conference on Software Engineering Companion*. 253–255.

[21] Google Developers. 2015. Real-Time Bidding Protocol: Cookie Matching. https://developers.google.com/ad-exchange/rtb/cookie-guide. (2015).

[22] Google Inc. 2011. "The Arrival of Real-Time Bidding and What it Means for Media Buyers". https://static.googleusercontent.com/media/www.google.com/en//doubleclick/pdfs/Google-White-Paper-The-Arrival-of-Real-Time-Bidding-July-2011.pdf. (2011).

[23] Wilson Andrews Gregor Aisch and josh keler. 2015. The Cost of Mobile Ads on 50 News Websites. http://www.nytimes.com/interactive/2015/10/01/business/cost-of-mobile-ads.html. (2015).

[24] Jiaping Gui, Ding Li, Mian Wan, and William G. J. Halfond. 2016. Lightweight Measurement and Estimation of Mobile Ad Energy Consumption. In *Proceedings of the 5th ACM International Workshop on Green and Sustainable Software*.

[25] Jiaping Gui, Stuart Mcilroy, Meiyappan Nagappan, and William G. J. Halfond. 2015. Truth in Advertising: The Hidden Cost of Mobile Ads for Software Developers. In *Proceedings of the 37th IEEE International Conference on Software Engineering - Volume 1*. 100–110.

[26] Patrick Holland. 2017. Verizon, T-Mobile, AT&T and Sprint unlimited plans compared. https://www.cnet.com/news/how-does-verizon-unlimited-plan-stack-up-against-the-others/. (2017).

[27] IAB Technology Laboratory. 2017. OpenRTB (Real-Time Bidding). https://www.iab.com/guidelines/real-time-bidding-rtb-project/. (2017).

[28] ImproveDigital. 2014. OpenRTB API Specification Document. http://www.improvedigital.com/main/wp-content/uploads/2014/05/OpenRTB22_Improve_Spec_final.pdf. (2014).

[29] Interactive Advertising Bureau (IAB). 2016. Rothenberg Says Ad Blocking Is a War against Diversity and Freedom of Expression. https://www.iab.com/news/rothenberg-says-ad-blocking-is-a-war-against-diversity-and-freedom-of-expression/. (2016).

[30] Azeem J. Khan, V. Subbaraju, Archan Misra, and Srinivasan Seshan. 2012. Mitigating the True Cost of Advertisement-supported "Free" Mobile Applications. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications (HotMobile)*.

[31] Nicolas Kourtellis, Jeremy Blackburn, Cristian Borcea, and Adriana Iamnitchi. 2015. Special Issue on Foundations of Social Computing: Enabling Social Applications via Decentralized Social Data Management. *ACM Transactions on Internet Technology* 15, 1 (2015).

[32] Nicolas Kourtellis, Joshua Finnis, Paul Anderson, Jeremy Blackburn, Cristian Borcea, and Adriana Iamnitchi. 2010. Prometheus: User-controlled P2P Social Data Management for Socially-aware Applications. In *Proceedings of the ACM/IFIP/USENIX 11th International Conference on Middleware*. Springer-Verlag, Berlin, Heidelberg, 212–231.

[33] Balachander Krishnamurthy and Craig Wills. 2009. Privacy Diffusion on the Web: A Longitudinal Perspective. In *Proceedings of the 18th ACM International Conference on World Wide Web (WWW)*. 541–550.

[34] Krux Digital Inc. 2010. Cookie Synching. http://www.krux.com/blog/ceos-corner/cookie-synching/. (2010).

[35] Rainey Reitman Kurt Opsahl. 2013. The Disconcerting Details: How Facebook Teams Up With Data Brokers to Show You Targeted Ads. https://www.eff.org/deeplinks/2013/04/disconcerting-details-how-facebook-teams-data-brokers-show-you-targeted-ads. (2013).

[36] David Martin, Hailin Wu, and Adil Alsaid. 2003. Hidden Surveillance by Web Sites: Web Bugs in Contemporary Use. *Commun. ACM* 46, 12 (Dec. 2003), 258–264.

[37] MoPub. 2016. MoPub OpenRTB 2.3 Integration Guide. https://dev.twitter.com/mopub-demand/marketplace-integration/openrtb. (2016).

[38] Brian Morrissey. 2015. Forbes starts blocking ad-block users. http://digiday.com/publishers/forbes-ad-blocking/. (2015).

[39] Richard Mortier, Jianxin Zhao, Jon Crowcroft, Liang Wang, Qi Li, Hamed Haddadi, Yousef Amar, Andy Crabtree, James Colley, Tom Lodge, Tosh Brown, Derek McAuley, and Chris Greenhalgh. 2016. Personal Data Management with the Databox: What's Inside the Box?. In *Proceedings of the ACM Workshop on Cloud-Assisted Networking*. 49–54.

[40] BH Murray and JJ Cowart. 2001. Web bugs: A study of the presence and growth rate of Web bugs on the Internet. Technical Report. (2001).

[41] David Naylor, Alessandro Finamore, Ilias Leontiadis, Yan Grunenberger, Marco Mellia, Maurizio Munafò, Konstantina Papagiannaki, and Peter Steenkiste. 2014. The Cost of the "S" in HTTPS. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies (CoNEXT)*. 133–140.

[42] Lukasz Olejnik, Minh-Dung Tran, and Claude Castelluccia. 2014. Selling off User Privacy at Auction. In *21st Annual Network and Distributed System Security Symposium, NDSS, San Diego, California, USA, February 23-26*.

[43] OpenX. 2016. RTB Macros. http://docs.openx.com/Content/demandpartners/rtb_macros.html. (2016).

[44] Elias P. Papadopoulos, Michalis Diamantaris, Panagiotis Papadopoulos, Thanasis Petsas, Sotiris Ioannidis, and Evangelos P. Markatos. 2017. The Long-Standing Privacy Debate: Mobile Websites vs Mobile Apps. In *Proceedings of the 26th ACM International Conference on World Wide Web (WWW)*. 153–162.

[45] Panagiotis Papadopoulos, Nicolas Kourtellis, Pablo Rodriguez Rodriguez, and Nikolaos Laoutaris. 2017. If You Are Not Paying for It, You Are the Product: How Much Do Advertisers Pay to Reach You?. In *Proceedings of the ACM Internet Measurement Conference (IMC)*. 142–156.

[46] Andrea Peterson. 2015. Bankrupt RadioShack wants to sell off user data. But the bigger risk is if a Facebook or Google goes bust. https://www.washingtonpost.com/news/the-switch/wp/2015/03/26/bankrupt-radioshack-wants-to-sell-off-user-data-but-the-bigger-risk-is-if-a-facebook-or-google-goes-bust/. (2015).

[47] PulsePoint. 2016. RTB Implementation Notes. http://docs.pulsepoint.com/display/BUYER/RTB+Implementation+Notes. (2016).

[48] RubiconProject. 2016. RTB API Documentation. http://dev.rubiconproject.com/docs/rtbapidocumentation. (2016).

[49] Khalid Saleh. 2016. Effectiveness Of Online AdvertisingâĂŞ Statistics And Trends. https://www.invespcro.com/blog/effectiveness-online-advertising/.

(2016).

[50] Sascha Segan. 2017. Verizon, AT&T May Be Choking Unlimited Data Users. https://www.pcmag.com/news/355963/verizon-at-t-may-be-choking-unlimited-data-users. (2017).

[51] Statista Inc. 2016. Premium Digital advertising spending worldwide from 2015 to 2020 (in billion U.S. dollars). https://www.statista.com/statistics/237974/online-advertising-spending-worldwide/. (2016).

[52] Narseo Vallina-Rodriguez, Jay Shah, Alessandro Finamore, Yan Grunenberger, Konstantina Papagiannaki, Hamed Haddadi, and Jon Crowcroft. 2012. Breaking for Commercials: Characterizing Mobile Advertising. In *Proceedings of the ACM Internet Measurement Conference (IMC)*. 343–356.

[53] Xuetao Wei, Lorenzo Gomez, Iulian Neamtiu, and Michalis Faloutsos. 2012. ProfileDroid: Multi-layer Profiling of Android Applications. In *Proceedings of the 18th ACM International Conference on Mobile Computing and Networking (Mobicom)*. 137–148.

[54] World Wide Web Consortium (W3C). 2010. Same Origin Policy. https://en.wikipedia.org/wiki/Same-origin_policy. (2010).

[55] Chanmin Yoon, Dongwon Kim, Wonwoo Jung, Chulkoo Kang, and Hojung Cha. 2012. AppScope: Application Energy Metering Framework for Android Smartphone Using Kernel Activity Monitoring. In *Presented as part of the USENIX Annual Technical Conference*. 387–400.

[56] Maciej Zawadzinski. 2018. How Does Real-Time Bidding (RTB) Work? https://clearcode.cc/blog/real-time-bidding/. (2018).

[57] John Zorabedian. 2016. Wired to adblocker users: pay up for ad-free site or you get nothing. https://nakedsecurity.sophos.com/2016/02/10/wired-to-ad-blocker-users-pay-up-for-ad-free-site-or-you-get-nothing/. (2016).