# Cookies That Give You Away: The Surveillance Implications of Web Tracking

Steven Englehardt
Princeton University
ste@princeton.edu

Dillon Reisman
Princeton University
dreisman@princeton.edu

Christian Eubank
Princeton University
cge@princeton.edu

Peter Zimmerman
Princeton University
peterz@princeton.edu

Jonathan Mayer
Stanford University
jmayer@stanford.edu

Arvind Narayanan
Princeton University
arvindn@princeton.edu

Edward W. Felten
Princeton University
felten@princeton.edu

## ABSTRACT

We study the ability of a passive eavesdropper to leverage "third-party" HTTP tracking cookies for mass surveillance. If two web pages embed the same tracker which tags the browser with a unique cookie, then the adversary can link visits to those pages from the same user (i.e., browser instance) even if the user's IP address varies. Further, many popular websites leak a logged-in user's identity to an eavesdropper in unencrypted traffic.

To evaluate the effectiveness of our attack, we introduce a methodology that combines web measurement and network measurement. Using OpenWPM, our web privacy measurement platform, we simulate users browsing the web and find that the adversary can reconstruct 62—73% of a typical user's browsing history. We then analyze the effect of the physical location of the wiretap as well as legal restrictions such as the NSA's "one-end foreign" rule. Using measurement units in various locations—Asia, Europe, and the United States—we show that foreign users are highly vulnerable to the NSA's dragnet surveillance due to the concentration of third-party trackers in the U.S. Finally, we find that some browser-based privacy tools mitigate the attack while others are largely ineffective.

## 1. INTRODUCTION

How much can an adversary learn about an average user by surveilling web traffic? This question is surprisingly tricky to answer accurately, as it depends on four things: the structure of the web, the mapping of web resources to the topology of the global Internet, the web browsing behavior of a typical user, and the technical capabilities and policy restrictions of the adversary. We introduce a methodology for quantifying the efficacy of passive surveillance. Our technique combines web measurement, network measurement, a client model (that incorporates user browsing behavior, web browser policies and settings, and privacy-protecting extensions), and an adversary model.

More specifically, the adversary has the ability to inspect packet contents and wishes to either track an individual target user or surveil users *en masse*. A key challenge for the adversary is the lack of persistent identifiers visible on the network (in Section 3 we discuss why IP addresses are inadequate). However, the adversary can observe HTTP cookies in transit. Indeed, both the NSA and GCHQ are known to use such cookies for surveillance (Section 3).

Our work starts with three insights. First, the presence of *multiple unrelated* third-party cookies on most web pages, albeit pseudonymous, can tie together most of a user's web traffic without having to rely on IP addresses (Figure 1). Thus the adversary can separate network traffic into clusters, with each cluster corresponding to only one user (or more precisely, one browser instance). A single user's traffic may span more than one cluster if the linking is imperfect.

Second, a significant portion of a typical user's traffic traverses U.S. borders even when the user is outside the U.S. and browses local content. As it turns out, such sites frequently include third-party resources such as analytics scripts, tracking pixels, and advertisements from U.S. servers. This leaves foreign users particularly vulnerable to the NSA's wiretaps within the U.S. under the "one-end foreign" rule (Section 3).

Third, although most popular websites now deploy HTTPS for authentication, many web pages reveal an already logged-in user's *identity* in plaintext. Thus, an adversary that can wiretap the network can not only cluster together the web pages visited by a user, but can then attach real-world identities to those clusters. This technique relies on nothing other than the network traffic itself for identifying targets.

Figure 1 illustrates the basis for our work. The adversary observes the user visit three different web pages which embed trackers $X$, $Y$ or both. The user's IP address may change between visits to each page, though we assume it is consistent for the request to site $A$ and the request to $A$'s embedded tracker $X$. But there is no way to tie together her visits to pages $A$ and $B$ until she visits $C$ after which all three visits can be connected. The unique cookie from $X$ connects $A$ and $C$ while the one from $Y$ connects $B$ and $C$. We assume here that the user has visited pages with both trackers before so that cookies have already been set in her browser and will be sent with each request. While IP address is a *convenient* method to link a request to a first party page to the corresponding request to an embedded third party tracker, it is not necessary. In Section 6.1 we show how this linkage can be achieved even if the IP address cannot be observed at all or if an IP address is shared by many users.

**Contributions.** Our contributions are both conceptual and empirical. First, we identify and formalize a new pri-


*WWW 2015,* May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3469-3/15/05.
http://dx.doi.org/10.1145/2736277.2741679.

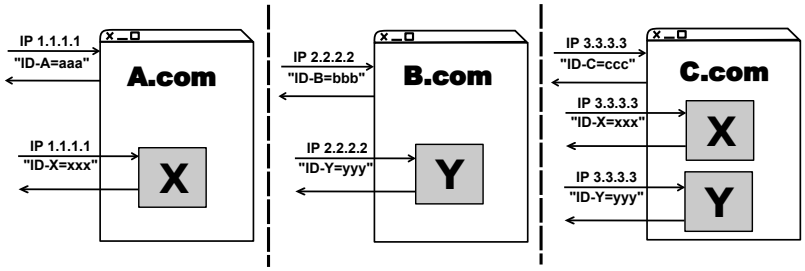



Figure 1: Illustration of link between each of a single browser's visits to three first-party pages using two different third-party tracking cookies. The user accesses the web at three different times, behind three different IP addresses.

vacy threat from packet sniffing. While the technique of utilizing cookies to target users is well known, we formulate the attack concretely in terms of the following steps: (1) automatically classifying cookies as unique identifiers (2) using multiple ID cookies to make *transitive* inferences and *clustering* HTTP traffic, (3) geographically tagging the flow of traffic, and (4) inferring real-world identity from HTTP request and response bodies. We believe this attack to be the strongest known way for a passive adversary to utilize web traffic for surveillance.

Second, we rigorously evaluate the above attack model using a novel methodology that combines web measurement, network measurement, a user model, and an adversary model (Section 4). We simulate realistic browsing behavior and measure the actual cookie ecosystem. This requires nuanced techniques along several fronts: (1) a crawling infrastructure based on *browser automation* to more closely simulate real users (2) a model of browsing history derived from real user behavior, and (3) network measurements to help verify the robustness of geolocation data.

Third, we present an exhaustive empirical evaluation of browser privacy settings and privacy extensions to discover how well a proactive user can protect himself against the attack model we've defined. Specifically, we measure how each of the following privacy measures affect the effectiveness of the attack: blocking all third-party cookies, blocking only those from sites not visited directly, setting Do Not Track (DNT), using Ghostery, and using HTTPS Everywhere.[1]

**Results.** We simulate users that make 0 to 300 web page visits spread out over a 2–3 month period. We consider users located in several possible countries. For each such set of visits, we perform clustering using the method described above and find the "giant connected component." For non-U.S. users, we consider an adversary with wiretaps in the target user's country as well as one with wiretaps in the U.S.

At a high level, our results show that for a U.S. user, over 73% of visits fall into this connected component. The clustering effect is extremely strong and is robust to differences in the models of browsing behavior. Clustering even occurs when the adversary is able to observe only a small, random subset of the user's requests.

Non-U.S. locations show a lower degree of clustering due to a lower number of embedded third-parties: over 59% of traffic falls into the giant connected component. If the adversary is further restricted to be in the U.S., the clustering level does drop significantly (12% – 20%), but this is still surprisingly high given that these users are browsing local content.

[1]DNT: `http://donottrack.us/`, Ghostery: `https://www.ghostery.com`, HTTPS-E: `https://www.eff.org/https-everywhere`

Second, we measure the presence of identifying information in plaintext among popular (Alexa Top 50 U.S.) websites. 56% of sites transmit some form of identifying information in plaintext once a user logs in, whether first name, first and last name, username, or email address. The majority of these (42% of websites overall) present *unique* identifiers (username or email address) in the clear.

Third, we show that many built-in browser protections are able to reduce but not fully mitigate the attack. The most effective blocking solution, Ghostery, still allows 24.2% of a user's traffic to be clustered, while alternative solutions have far less of an impact.

**Implications** An adversary interested in targeted surveillance can proceed as follows: (1) Either scan for the target's identity in plaintext HTTP traffic, or use auxiliary methods to obtain the target's cookie ID on some first-party page (2) From this starting point, transitively connect the target's known first-party cookie to other third-party and first-party cookies of the target. On the other hand, an adversary interested in *en masse* surveillance can first cluster all observed HTTP traffic, albeit at a high computational cost, and then attach identities to these clusters using the methods above. Our attacks show the feasibility of either adversary's goal.

What can the adversary do after attaching an identity to a cluster of web traffic? First, browsing history itself could be the information of interest, as in the NSA plan to discredit 'radicals' based on browsing behavior, such as pornography [4]. Second, further sensitive information might be found in unencrypted web content such as preferences, purchase history, address, etc. Finally, it can enable active attacks such as delivering malware to a targeted user [47, 19].

## 2. RELATED WORK

Our work draws from two previously independent bodies of research. The first analyzes the privacy implications of third-party cookies and the second analyzes the ability of a network eavesdropper to infer sensitive information. We describe each in turn. To our knowledge, the two sets of ideas have not been combined before.

**Third-party tracking: prevalence and privacy implications.** There have been several notable results uncovering or quantifying various types of online tracking: cookies [28, 45], flash cookies (LSOs) including respawning behavior [46, 9, 38, 6], and browser fingerprinting [41, 7, 6]. The ability of trackers to compile information about users is further aided by PII leaks from first parties to third parties [29, 30, 27] and "cookie syncing", or different third-party trackers linking their pseudonymous cookies to each other [42, 6].

While this body of research helps us understand what trackers themselves can learn, it does not address the question we are interested in, which is what an *eavesdropper* can

learn through cookies and identifier leaks. The latter is influenced by many additional factors including geographic location of the trackers and the adoption of HTTPS by websites.

Yen et al. show how IP, cookies and usernames can be combined to track devices reliably even when any one of these identifiers may be individually unreliable [50]. The similarities to our work are superficial: we study linkage by an eavesdropper who utilizes third-party cookies rather than a website that uses its first-party cookies. The goals are also different: learning users' web histories vs. ID-ing devices.

There are various client-side tools to block, limit or visualize third-party tracking. These are too numerous to list exhaustively, but a sampling include Adblock Plus, Ghostery, ShareMeNot [1], Lightbeam, and TrackingObserver [2]. Studies that have quantified the privacy effect of these tools include [35, 11, 20].

**Surveillance: attacks, defenses, and measurement.** While there is a large body of work on what a passive adversary can infer about users, virtually all of it concerns attacks arising from side-channels in encrypted traffic, particularly Tor. While Tor is insecure against a global passive adversary, traffic analysis attacks have been studied with respect to passive and active adversaries with less comprehensive access to the network [39, 40]. *Website fingerprinting* allows a local eavesdropper to determine which of a set of web pages the user is visiting, even if the connection is encrypted, by observing packet lengths and other features [25, 24, 43]. Other side-channel attacks include timing attacks on SSH [48], leaks in web forms, [15] and inferring spoken phrases from VoIP [49].

By contrast, we study users who do *not* use a properly configured Tor browser. The adversary's main challenge is not linking origin and destination, but rather linking different traffic flows to the same (real-world) identity.

Closer to our work, Arnbak and Goldberg studied how the NSA could actively redirect U.S. traffic abroad, so as to bring it within broader surveillance authorities [8].[2] The IXMaps tool allows users to interactively view the routes taken by their traffic and intersection with known NSA wiretapping sites [16].

# 3. BACKGROUND AND THREAT MODEL

In recent years, a combination of technical research, leaks, and declassifications has provided unprecedented transparency into Internet surveillance by governments. Some nations, such as Iran and Bahrain [17], practice near-total Internet monitoring. Others, including the United States and Britain, have large-scale technical capacity—but subject to legal limits. This section explains how the National Security Agency (NSA) and Government Communication Headquarters (GCHQ) have used third-party cookies in their respective surveillance programs, as well as briefly discusses the laws that govern surveillance of Internet traffic within the United States. It then sets out a threat model that motivates our study.

**NSA and GCHQ use of third-party cookies.** Leaked documents reflect at least three ways in which the NSA has used third-party cookies obtained from its Internet intercepts. First, the agency has investigated passively identifying Tor users by associating cookies with non-Tor sessions. Specifically, the NSA attempted to link a Google third-party advertising cookie between Tor and non-Tor sessions [5].

Second, the agency has an active, man-in-the-middle system ("QUANTUMCOOKIE") that induces cookie disclosure [5]. Applications include identifying Tor users and targeting malware delivery.

Third, the agency has used passively obtained cookies to target active man-in-the-middle exploitation. On at least one occasion, the NSA offered a Google cookie to single out a user for exploitation [47].

In addition to these specific applications, HTTP analytical tools (such as "XKEYSCORE") incorporate cookie data. An analyst could easily take advantage of third-party cookies when querying intercepted data [21].

Several leaked documents also reveal two GCHQ programs for surveilling and targeting users via third-party tracking data, both from web browsers and mobile applications. One program, "MUTANT BROTH", a repository of tracking cookies linked with additional metadata such as IP addresses and User-Agent strings. This repository is reported to have been used for targeted malware delivery [19].

The other program, "BADASS", offers a similar repository and search interface for querying information leakage from mobile apps. The system collects and extracts leaked identifiers, device and operating system details, and additional information transmitted in plaintext [31].

**United States Internet monitoring.** The law surrounding NSA authority derives from a complex mixture of constitutional doctrine, statutory restrictions, and executive regulation. One emergent property is that, when at least one party to a communication is outside the United States, it is eligible for warrantless surveillance.[3] "Upstream" interception devices, controlled by the NSA and foreign partners, are exposed to large volumes of this "one-end foreign" Internet traffic. While the details remain classified, it also appears that a substantial quantity of one-end foreign traffic is temporarily retained. Leaks indicate that at least some installations of "XKEYSCORE," a distributed analysis system, maintain a multi-day buffer of Internet traffic [12].

**Threat model.** In developing a threat model, there are two extremes, neither of which is desirable. The first is to assume that the adversary is all-powerful, as in cryptographic security arguments. Such a model is both uninteresting and largely irrelevant to the real world. The other extreme is to focus too closely on the NSA or GCHQ's activities. Such a model may not yield insights that apply to other surveillance programs and the results may be invalidated by changes to the respective agency's programs. We seek a careful middle ground and arrive at a model that we believe is realistic enough to influence policy debates and privacy tool development, yet general enough for our analyses and algorithms to be of independent scientific interest and for our results to hold up well over time.

We consider only passive attacks for several reasons. First, passive attacks appear to be more powerful than generally

[2]It is not apparent whether the NSA has redirected traffic in this manner, nor is it apparent whether the agency would consider the practice lawful.

[3]One-end foreign wireline interceptions inside the United States are generally governed by Section 702 of the FISA Amendments Act [22]. Two-end foreign interceptions and one-end foreign wireless interceptions inside the United States are generally governed by Executive Order 12333. Interceptions outside the United States are also generally governed by Executive Order 12333 [3].

realized, and we wish to highlight this fact. Second, even an active attack must usually begin with passive eavesdropping. An adversary must have refined criteria for targeting the active attack. Finally, almost all active attacks carry some risk of detection. Passive attacks much easier to mount, especially at large scale.

We consider a powerful adversary with the ability to observe a substantial portion of web traffic on the Internet backbone. The adversary's view of a given user's traffic may be complete or partial. We model partial coverage in two ways: by assuming that a random subset of the user's HTTP requests and responses flows through one of the adversary's wiretaps, or that the adversary taps the portion of the user's traffic that traverses United States borders. While the NSA has many interception points outside U.S. borders as well, the U.S.-only model provides a useful, approximate lower bound of the agency's abilities. We also assume that the adversary cannot routinely compromise HTTPS, so cookies or other identifying information sent over HTTPS are of no use.

The adversary may have one of two goals: first, he might want to target a specific individual for surveillance. In this case the adversary knows either the target's real-world identity or a single ID cookie known to belong to the target (whether on a first or third party domain). Second, the adversary might be engaged in mass surveillance. This adversary would like to "scoop up" web traffic and automatically associate real-world identities with as much of it as possible.

The adversary's task is complicated by the fact that the IP addresses of the target(s) may change frequently. A user's IP address could change because she is physically mobile, her ISP assigns IP addresses dynamically, or she is using Tor. Leaked GCHQ documents show that their search interface even warns analysts to take care when selecting data on dynamic IPs [19]. Browsing from a smartphone is a case worth highlighting: Balakrishnan et al. find that "individual cell phones can expose different IP addresses to servers within time spans of a few minutes" and that "cell phone IP addresses do not embed geographical information at reasonable fidelity" [10].

To link users across different networks and over time, the adversary aims to utilize first-party and third-party unique cookies assigned to browser instances by websites. He can easily sniff these on the network by observing the "Cookie" field in HTTP request headers and the "Set-Cookie" field in HTTP response headers. Cookies set by an "origin" (roughly, a domain) that have not expired are automatically sent as part of requests to the same origin.

# 4. METHODOLOGY

In this study, we wish to simulate real users browsing over a period of time, detect the creation of unique identifiers, and measure the flow of both unique pseudonymous and real-world identifiers to adversaries with differing collection capabilities. We present a summary of our methodology below, and provide detailed descriptions of our measurement and analysis methodology in the following subsections.

1. Define all clients and adversaries to be studied, according to the following models:
   - client: (location, browsing model, browser configuration) which encodes the client's geographic and network location, which sites the client visits, and the browser settings and plugins the client browses with.
   - adversary: (location, policy restrictions) which encodes the adversary's geographic and network location, and the policy restrictions on data use and collection.
2. For each unique (user location, browsing model) pair of interest, generate $N$ simulated browsing profiles as defined in Section 4.1 and create a corresponding client instance for each one.
3. Use the measurement infrastructure (Section 4.2) to simulate the users defined in Step 2 and collect all network traffic (i.e. HTTP requests, responses, and cookies). Our measurements are summarized in Section 4.3.
4. For each (client location, web resource) pair of interest, determine the geographic path of traffic using the procedure described in Section 4.4.
5. Run the ID cookie detection algorithm detailed in Section 4.5 to flag identifying cookies
6. For each (client, adversary) pair of interest, do the following for all instances of the client and average the results:
   - filter the list of requests based on the geographic location and policy restrictions of the adversary using the geographic mapping created in Step 4.
   - run the cookie linking algorithm detailed in Section 4.6 using the ID cookies detected in Step 5.
   - report the size of the connected components in the linking graph (as a ratio of total number of visits)
   - report the number of sites known to leak real-world identifiers (Section 4.7) contained in each component.

## 4.1 Browsing models

We use two browsing models to create simulated user profiles. One of our models was a naive one – the user visits random subsets of the Alexa top 500 sites local to the location of the measurement instance. For example, a measurement instance in Japan would sample the Alexa top-500 sites for users in Japan, while a measurement instance in Ireland would sample from the Alexa top-500 sites for users in Ireland.

Our other browsing model aims for realism by making use of the AOL search query log dataset. The dataset contains queries made by 650,000 anonymous users over a three month period (March–May 2006). We create a browsing profile from a user's search queries as follows. First, we remove repeated queries. Next, for every search query performed by the user, we submit the query to Google search and retrieve the links for the first five results. Users were selected on the basis that they performed between 50 to 100 unique queries which resulted in browsing profiles of 250 to 500 URLs. This is almost identical to the method that was used in [32].

Of course, only a subset of real users' web browsing results from web searches. Nevertheless, we hypothesize that our profiles model two important aspects of real browsing histories: the distribution of popularity of web pages visited, and the topical interest distribution of real users. Popular websites may embed more trackers on average than less popular sites, and websites on the same topic may be more interconnected in terms of common embedded trackers. Failure to model these aspects correctly could skew our results.

The reason we recreate the users' searches on a current search engine rather than simply using the sites visited by the AOL users (available in the dataset) is that the distribution of websites visited by real users changes over time as websites rise/fade in popularity, whereas the distribution of users' interests can be expected to be more stable over time.

## 4.2 Measurement infrastructure

We built our study on top of a web measurement platform, OpenWPM [18], which we developed in earlier work. The platform has the ability to drive full browsers, such as Firefox, with any set of browser extensions and collects a wide range of measurement data as it visits sites.[4] In our study, we use version 0.1.0 of OpenWPM to drive Firefox measurement instances from which we record all HTTP data for analysis. We configure the measurement instances to browse with profiles generated from the models described in Section 4.1, and deploy the crawls on cloud machines in the United States, Japan, and Ireland.

## 4.3 Measurements

We deployed OpenWPM on Amazon EC2[5] instances in three regions: Northern Virginia, United States, Dublin, Ireland, and Tokyo, Japan. We chose these to achieve as much geographic diversity as possible from Amazon's limited set of regions. Each measurement took place on an m3.medium instance of Ubuntu 14.04 in June 2014. All measurements were ran using 25 simulated profiles for each (user location, browsing model, browser configuration) combination.

When making measurements from within the U.S., we were able to utilize the more realistic AOL browsing model. We used it under serveral browser configurations: no cookie blocking, blocking third-party cookies from sites which the user has not yet visited as a first-party, blocking all third-party cookies, setting the DNT flag, browsing with HTTPS Everywhere installed, and browsing with Ghostery installed and configured to block all possible entities.

For measurements outside of the Unites States, we were not able to utilize the AOL browsing model as the search terms and results are likely biased towards U.S. users. Instead, we fall back to the Alexa browsing model when doing comparisons between geographic locations. To compare measurements between the United States, Japan, and Ireland we used an Alexa browsing model localized to the most popular sites within each country.

To run the identifying cookie detection algorithm described in Section 4.5, we also require synchronized measurements of each site visit from two separate machines. We ran these measurements from the Northern Virginia location and visited all of the links which may be visited by any other measurement instance (13,644 links total).

For all measurements, web pages were visited approximately once every ten seconds. We set a 60 second timeout per visit and restarted the browser with consistent state in the event of a crash.

## 4.4 HTTP Traffic geolocation

In order to determine the if an HTTP request is bound for a specific location of interest, we augment commercially available geolocation data with additional measurement data. After each measurement instance finished browsing, we ran a `traceroute`[6] to each unique hostname and recorded the full output. All IPs returned in each hop of the traceroute were geo-located with the `MaxMind GeoLite2`[7] country databases.

The mapping between IP and physical location is not one-to-one. Instead, there may be many servers in different locations which all share the same IP address for various purposes. One such example is *anycasting*, the process by which several nodes share the same address and the user is routed to the nearest node.[8]

Thus, when determining if an HTTP request enters a specific country it is not sufficient to simply geolocate the IPs returned from a traceroute to that host. As a solution, we implement a simplified version of the *geo-inconsistency* check proposed by Madory, et.al. [33]. We check that the minimum round-trip time (RTT) returned by a traceroute to each hop is greater than the physical minimum RTT assuming a direct fiber-optic connection between the two locations.

Algorithm 1 summarizes the steps we take to perform this origin-specific geolocation check. Broadly, if the geolocation of a specific hop returns as being within the country of interest, we find the travel time between the (latitude, longitude) pairs of the origin server and the geolocated IP. If geolocated IP's location is on the country level, we choose the closest point in the geolocated country from the origin location. We then use the haversine forumla to calculate the distance between the two points and find the minimum RTT:

$$minRTT = 2 * \frac{\text{haversine distance} * n}{c}$$

where $c$ is the speed of light in units matching the distance measurement and $n$ is the optical index of the fiber. In our study, we use $n = 1.52$ as the reference optical fiber index.

```
Data: httpRequest, testCountry
Result: True/False if httpRequest enters testCountry
origin ← latitude/longitude of origin server
hostname ← parse httpRequest.url
for each hop in hostname traceroute do
    location ← geolocate(hop.IP)
    if location not in testCountry then
        continue
    if location not city code then
        location ← closest point to origin within country
    dist ← haversine(origin,location)
    minRTT ← 2 * minimum time to travel dist
    if hop.RTT > minRTT then
        return True
end
return False
```

**Algorithm 1:** Origin-specific geolocation check

This check does not guarantee that a specific request enters a country, as network delays could artificially push a traceroute RTT above the threshold. Our assumption of a straight-line connection and optical fiber index is also unlikely to hold in practice. Instead, this check provides a more realistic upper-bound on the amount of traffic an adversary at a specific geographic location can monitor. For example, this check eliminated the numerous examples we observed of traceroutes originating in Ireland and Japan having geolocations within the United States with RTTs of $<5$ms.

We use a simplified version of this check when examining if requests are exiting the United States. Since a request can

[4] `https://github.com/citp/OpenWPM`

[5] `https://aws.amazon.com/ec2/`

[6] Our traceroutes were configured to use a single probe per hop with a maximum of 25 hops.

[7] `http://dev.maxmind.com/geoip/geoip2/geolite2/`

[8] CloudFlare, for example, claims to use anycasting as part of their content delivery network: `https://www.cloudflare.com/features-cdn`

be bound for any non-U.S. destination, we do not attempt to find the closest point in each country. Instead, we only check that the observed RTT is greater than the minimum RTT to the geolocation point regardless of the point's location.

## 4.5 Detecting unique identifier cookies

An essential task to quantifying our attack is the ability to detect which cookies are identifying. Identifiers can be stored in many locations (e.g. HTTP cookies, Flash cookies), but to be sent back to trackers the identifiers must be included in HTTP cookies or query parameters of the request. We choose to focus HTTP cookies as they are included in every request and thus provide a generic approach that does not necessitate the parsing of URL parameters for all sites under surveillance. Furthermore, non-cookie tracking techniques are ordinarily paired with tracking cookies; our approach indirectly incorporates an adversary's capabilities against those technologies.

For our analysis, we are interested in the host (the domain that set the cookie), name, and value fields of a cookie and determine if the data stored in the value field is identifying. This algorithm is a modified version of one we used in a previous study [6].

To be useful to the adversary as identifiers, cookie values must have two important properties: persistence over time and uniqueness across different browser instances. Based on these criteria we develop an algorithm that classifies cookies as identifiers. Our algorithm is intentionally conservative, since false positives risk exaggerating the severity of the attack. Our method does have some false negatives, but this is acceptable since it is in line with our goal of establishing lower bounds for the feasibility of the attack.

We define a cookie to be an identifier cookie if it meets the following criteria: (1) It is *long-lived*, with expiry times longer than three months. The three month cut-off matches our AOL dataset three month window. (2) It's value is *stable*, and remains constant through all page visits. Dynamic value strings may be timestamps or other non-identifiers. (3) Has a *constant-length* across all our datasets. (4) Is *user-specific*, so the values are unique across different browser instances in our dataset. (5) Has a *high-entropy* value string, with values sufficiently different between machines to enable unique identification. To test for this, we used the Ratcliff-Obershelp [13] algorithm to compute similarity scores between value strings. We filtered out all cookies with value strings that were more than 55% similar to value strings from the corresponding cookies of different measurements.

We run our detection algorithm on the synchronized measurement data described in Section 4.3. By using synchronized measurements, we avoid the problem of sites changing their cookie interaction behavior depending on a user's browsing time. For instance, in relation to the entropy heuristic, cookies with values that depend on time stamps will be easier to detect and ignore if the crawls have nearly the same timestamps for all actions. For other measurements, we extract identifying cookie values by searching for cookies which match the identifying (host, name) pairs classified in the synchronized measurements.

## 4.6 Transitive Cookie Linking

**Building the graph.** Once we determine which cookies contain unique identifiers, we use the `http_requests`, `http_responses`, and `http_cookies` tables of the OpenWPM crawl database to cluster traffic. From these tables, we construct cookie linking graph using Algorithm 2, which creates a graph with two node types: *URL* nodes and *Cookie* nodes. URL nodes are identified by the tuple `(U, <node url>, <request's geographic destination>)` and cookie nodes consisting of the tuple `(C, <cookie_value>)`.

Edges are created under the assumption that a network adversary will be able to link all requests and responses for a single page visit together if he can both follow the chain of referrer and redirect headers for HTTP requests from a single IP. URL — URL edges are created under two conditions: (1) one url node was observed as the `referer` on a request to the connected url node or (2) one url node was returned in the location field of a `301` or `302` redirect response to the request for the connected url. An adversary is only able to link together different page visits by the shared cookie values loaded on each page. As such, URL — Cookie edges are created whenever a cookie value is observed in the `Cookie` field of an HTTP Request header or the `Set-Cookie` field of an HTTP Response header. Notice that the only linking between separate page visits in the graph occurs when two HTTP requests/responses happen to link to the same Cookie node, while referrer and redirection chaining provides linking within a page visit.

**Data**: httpRequests and httpResponses for $user_i$
**Result**: Graph $G_i$ for $user_i$ with $URL$ & $Cookie$ nodes
**for** *each visited url* **do**
  **for** *httpRequest* **do**
    **if** *HTTPS* **then**
      **continue**
    urlNode ← createNode (U, req.url, req.inUS)
    G.addNode(urlNode)
    **if** *req.referer is not empty* **then**
      refNode ← createNode (U, ref.url)
      G.addNode(refNode)
      G.addEdge(refNode, urlNode, req.inUS)
    **if** *req.cookie is not empty and is identifying* **then**
      cookieNode ← createNode (C, cookie.value)
      G.addNode(cookieNode)
      G.addEdge(cookieNode, urlNode, req.inUS)
  **end**
  **for** *httpResponse with Set-Cookie* **do**
    **if** *HTTPS* **then**
      **continue**
    **if** *cookie is identifying* **then**
      cookieNode ← createNode (C, cookie.value)
      urlNode ← node for requested url
      G.addNode(cookieNode)
      G.addEdge(cookieNode, urlNode, req.inUS)
  **end**
  **for** *httpResponse with location field* **do**
    **if** *HTTPS* **then**
      **continue**
    urlNode ← node for requested url
    locNode ← createNode (U, loc.url)
    G.addNode(locNode)
    G.addEdge(locNode, urlNode, req.inUS)
  **end**
**end**

**Algorithm 2:** Cookie Linking algorithm

**Analyzing the graph.** In our analysis all graphs only contain traffic for a single user. This allows us to find the connected components within the graph and utilize the giant connected component (GCC) to find the amount of a single user's traffic an adversary is able to link. Once the GCC is found, we take the intersection of the set of URLs contained in the GCC with the set of URLs visited during the measurement to find the amount of top-level page visits an adversary is able to observe. When the adversary applies the attack in a multi-user setting, they will have many disjoint subgraphs per user of varying size. Depending on the adversary's goal, these clusters could be processed to link them individually to real world identities, or disambiguated by identifying disjoint sets of cookies for the same sites.

When evaluating adversaries restricted by policy or geographic constraints, an additional pre-processing step is required before finding the GCC. Utilizing the geolocation data from Section 4.4, we are able to filter nodes from the cookie linking graph based on geographic restrictions. In order to determine the amount of traffic for a specific user that a U.S. restricted adversary has access to, we filter all edges from the cookie linking graph that were not added due to U.S. bound requests. We create a subgraph from this filtered edge list and continue the normal linking analysis.

## 4.7 Identity leakage in popular websites

We conducted a manual analysis of identity leaks on the most popular pages which allow account creation. The top-50 pages are a useful representation for how heavily-used sites with user accounts manage data, and are more likely to be visited by a real user. We identified 50 of the Alexa top 68 U.S. sites that allow for account creation and signed up test accounts when possible. We then examined the homepage, account page, and several random pages on each site to see if any of identifiers are displayed on an HTTP page. If so, an adversary collecting HTTP traffic for that user could inspect the contents of the page to find the identifier and link it to any tracking identifiers also loaded on the page.

# 5. RESULTS

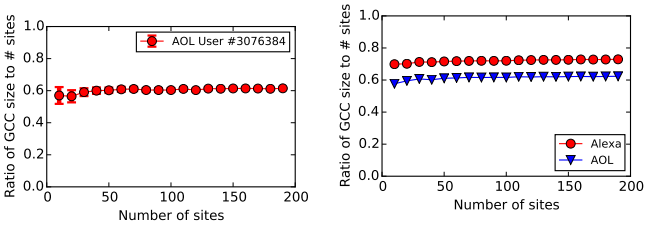
In the course of our measurements we make nearly 100,000 page visits to 13,644 distinct sites under several client and adversary models. Of these 13,644 sites, nearly all of them (13,266) make requests for external content from a host different than the domain of the host visited.

## 5.1 Clustering

In this section, we evaluate the effectiveness of our proposed attack under several (adversary, client) models. We are primarily interested in the number of web pages visited by a user which are located in the giant connected component (GCC) relative to the number of pages with embedded third-parties. We focus on large connected components because the probability that a cluster will have at least one page visit that transmits the user's real-world identity in plaintext increases with the size of the cluster. Manual inspection shows that page visits not in the large connected components belong to small clusters, typically singletons, and are thus unlikely to be useful to the adversary.

**An adversary's incomplete view.** We must consider that the adversary's position on the network may not give them a full view of any user's traffic. The adversary's view may be limited due to its policy or geographic restrictions as described in Section 3, however even with these considerations an adversary may not see all of a user's traffic. A user may change locations on the network or alternative routes taken by packets through the network might result in gaps in the adversary's data collection. To model the adversary's partial view of the user's browsing activity, we repeat our analysis with random subsets of web pages of various sizes.

For illustration, Figure 2a shows how the GCC of a single AOL user's page visits ($y$-axis) grows as we vary the completeness of the adversary's view of the user's HTTP traffic ($x$-axis). Each data point was computed by taking 50 independently random samples of page visits. For each sample we apply the clustering algorithm and compute the fraction contained in the GCC. We then average the fractions across the 50 samples. Since we wish to simulate these page visits being spread out over time, only cookies with expiration times at least three months into the future were included when computing the clusters.




(a) Individual AOL User (b) Average AOL & Alexa Users

Figure 2: Clustering, random subsets of traffic

Thus for each $(x, y)$ pair we can say that if the adversary captures $x$ web page visits by a user in the course of a wiretap, they could link approximately $y$% of those visits into a single cluster. The numbers we see for this user are typical — the fraction is around 55% for even very small clusters and exceeds 60% as the cluster size increases. As discussed in Section 4, we average all results over $N = 25$ client instances for all (client, adversary combinations).

We alternatively examined an adversary who sees subsets of web pages that the user visited in chronological order (perhaps the adversary only observed the user for a short period of time). These results had no statistically significant differences from random subsets. As such, we present the remainder of the graphs using random subsets.

**Unrestricted Adversary — AOL User Profiles** We first examine the level of clustering an adversary can achieve when it is not subject to any policy or geographic restriction. The results for users simulated under the AOL browsing model and no blocking tools are included in Figure 2b. These results show that the clustering remains very similar to the single user case of Figure 2a, except that no inconsistencies remain. After just 55 web page visits observed by the adversary, the growth of the GCC flattens out to 62.4±3.2% after 200 sites. For the remaining results, we will only present values for this asymptotic case of 200 sites, as the shape of the GCC growth is nearly identical in all cases.

**Unrestricted Adversary — Alexa profiles** Next we examine the effect of the browser model on the ability of an unrestricted adversary to cluster user data. We hold the user location and browser configuration set to browsing within the U.S. with no blocking settings. Figure 2b compares the results for Alexa profiles for U.S. users against the AOL profiles. The Alexa clustering shows a similar growth

pattern with an offset around 10% higher on average, with an overall level of clustering after two sites of 72.9±1%.

## 5.2 U.S. Users Under One-End Foreign

We now consider an adversary located within the United States who is constrained by the "one-end foreign" rule when collecting data on U.S. users. The client model used in evaluating this adversary is U.S.-based users browsing random subsets of the Alexa top-500 U.S. sites with no blocking tools. The size of the largest cluster observed reduces to just 0.9±0.2% of visited sites averaged across all instances.

To understand why this occurs, we must look at the composition of HTTP traffic. For the average user in this (adversary, client) pair, at least one non-U.S. sub-resource request occurs for 31.7% of the Alexa top-500 sites in the U.S. However, the overall number of HTTP Requests leaving the United States is comparatively small, accounting for just 2.0% of all requests. Only considering traffic where an adversary could learn the top-level domain through the referrer headers, this reduces to 22.7% of visits and 1.6% of requests.[9] Although nearly a quarter of a user's browsing history is visible to the adversary, we show that cookie linking is an ineffective method to cluster this traffic.

## 5.3 Cookie Linking in Non-U.S. Traffic

We now explore the level of clustering that is possible for traffic with a non-U.S. origin. We examine two different cases in this section: we first show the level of clustering possible under the assumption that the adversary will see all web requests that occur for a specific page visit and then we show what an adversary observing U.S.-bound traffic would see. A key point to note is that even if the majority of a website's resources are requested from a local server, embedded third-parties may cause U.S.-bound requests to occur which have the domain of the first-party as a referrer.

| User Location | Unrestricted Adver. | US-based Adver. |
|---|---|---|
| Japan | 59.6±1.2% | 20.9±0.7% |
| Ireland | 63.8±1.2% | 12.8±1.1% |

Table 1: Clustering of non-U.S. users by an adversary with no restrictions vs. one restricted to U.S. bound traffic.

**Unrestricted Adversary — non-U.S. profiles** When all requests are considered, the level of clustering is similar to that of the U.S.-based Alexa user simulations. Table 1 shows amount of clustering that occurs for users in Ireland and Japan under the random subset clustering model. Simulated users in Ireland can expect around 63% of traffic to be clustered, while users in Japan can expect nearly 60%. We believe the differences between user simulations generated using Alexa's top U.S., Japan, and Ireland lists arises from the difference in the number of included third parties on the first party (i.e., 62, 38, and 30 on average, respectively).

**U.S. based Adversary — non-U.S. profiles** We then restrict the clustering to only consider requests which are U.S. bound, and cluster based on the information available to a geographically restricted adversary. This could include an adversary within the United States, or an adversary sitting at the entrance to undersea cables returning to the United States. Table 1 shows clustering capabilities of an adversary restricted to these conditions. In Ireland, we see a reduction to around 13% of page visits and in Japan we see a less severe decrease to 20% of visited sites.

## 5.4 Cookie Linking Under Blocking Tools

We now investigate several ways users may mitigate a clustering attacking using currently available consumer tools. For all blocking configurations, we make measurements using the AOL browsing model and we examine the ability of an unrestricted adversary to cluster traffic. Measurements are run using several different privacy settings within the browser and two popular privacy and security add-ons.

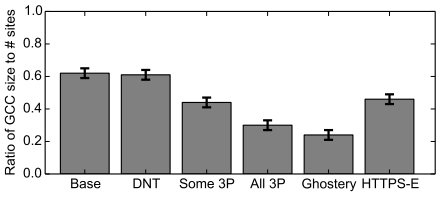



Figure 3: Clustering under several privacy settings.

*Baseline* displays the level of clustering with no privacy settings or blocking tools enabled. This represents an upper bound on the level of tracking we would expect to see under the other configurations.

*DNT* had a negligible effect on clustering, showing no statistically significant difference than without blocking.

*Cookie blocking* policies are more effective, particularly when an adversary sees a low number of page visits. We block cookies for sites that have not been visited in a first-party context by setting "Accept third-party cookies: From visited" in Firefox (this is also the default in Safari). When set, the level of clustering reduces to 43.9±3.2%. Blocking all third-party cookies further reduces this to 30.2±3.0%.

*HTTPS Everywhere* is an extension created by the EFF to force HTTPS connections whenever available. Since HTTPS requests are not visible to an attacker, and HTTP requests from HTTPS origins will not include a *referer* (preventing the attacker from linking requests back the the original site). A site can fall into one of four categories: no support for HTTPS, supported but not the default, supported and used by default, and finally, HTTPS-only. This measurement provides a picture of what happens as more sites support and use HTTPS by default. Under our browsing model, the number of HTTPS requests increases from 12.7% to 34.0% and the level of clustering is reduced to 46.1±3.2%.

*Ghostery* is a popular list-based browser extension for blocking third-party requests to domains considered to be trackers. This proves to be the most effective solution for users, reducing the level of clustering to 24.2±2.8% of visited sites. Enabling Ghostery and configuring it to block as much as possible reduces the average number of inclusions from external hosts to just 5.2 per first-party.

## 5.5 Identity Leakage

Table 2 summarizes our results from a manual survey of the Alexa U.S. sites. We picked the top 50 sites that support account creation. 44 of the 50 websites used HTTPS to secure login pages.[10] Only 19 of those sites continued to

[9]These results are broadly consistent with measurements taken by several of this paper's authors in past work [36].

[10]5 of the remaining 6 made POSTs with credentials and 1 made a GET with the credentials as a URL query parameter

use HTTPS to secure future interactions with the user after logged in. We summarize the cleartext identity leaks for the websites which no longer continue to use HTTPS after login.

Although a majority of sites secure user credentials on login pages, personally identifying information (name, username, email address) is transmitted much more frequently via HTTP. Over half of the surveyed sites leak at least one type of identifier, and 42% (not shown in table) leak either username or email address, which can be used to uniquely infer the user's real-world identity. Past work [34, 29] has found a roughly equivalent level of leakage to occur through the Request-URI and Referer.

A representative example of the web's identity-leak problem is `imdb.com`. IMDB provides a secure login page, but once logged in, users return to an HTTP page. This page includes the user's full name on the homepage of the site. Every time a user visits the site while logged in, a passive attacker can extract the name from the unencrypted traffic.

| Plaintext Leak Type | Percentage of Sites |
|---|---|
| First Name | 28% |
| Full Name | 14% |
| Username | 36% |
| Email Address | 18% |
| At least one of the above | 56% |

Table 2: Leakage on Alexa Top 50 supporting user accounts

Furthermore, we verified that pages from these popular sites that leak identity occur in the clusters of web pages found in our attack. Specifically, at least 5 (and an average 9.92) of the 28 sites we found to leak some type of identity were found in the giant connected component of every one of the 25 Alexa U.S. users. Due to global popularity of the top-50 U.S. sites, an average of 4.4 and 6.4 of these identity leaking sites are found in the GCC's of the Alexa Japan and Alexa Ireland users, respectively. Additionally, for the AOL profiles with no blocking, 9 of the 25 simulated users had at least 1 identity leaker in the GCC. Of course, there are likely also many sites outside the top 50 that leak identity and are found in these clusters, but we did not measure these.

Taken together with our results on clustering, our measurements show that a passive attack is highly feasible: after observing only a fraction of a user's web traffic the adversary will be able to link the majority of the user's web requests together and furthermore, use the contents of the responses to infer the user's real-world identity.

# 6. DISCUSSION

## 6.1 Linking without IP address

So far we have assumed that the adversary sees the same source IP address on a request to a first-party site and its corresponding third-party tracker, and that this can be used to link the two requests. There are at least two scenarios in which this assumption is problematic. The first is a NAT. If two users, Alice and Bob, behind the same NAT visit the same web page at roughly the same time, the adversary sees the same IP address on all ensuing HTTP requests. The other scenario is when the user employs Tor without proper application layer anonymization, and the adversary is able to sniff cookies only on the path from the exit node to the web server. (If the user is using a properly configured Tor setup, such as the Tor browser bundle, this attack does not work at all). Since Tor will, in general, use different circuits for communicating with different servers, the adversary will see different source IPs for the two requests (or may be able to observe only one of the requests).

However the well-known "intersection attack" can be used to link requests without using the IP address: if a cookie value $a$ associated with page A's domain and a cookie value $x$ associated with an embedded tracker domain $X$ are observed *multiple times* near-simultaneously (e.g. within 1 second of each other), then $a$ and $x$ are probably associated with the same user. Intuition suggests that for all but the busiest of web pages, two or three visits may be sufficient to link the first-party and tracker cookies with each other. However, this claim cannot be rigorously evaluated without access to large-scale HTTP traffic and so we leave this as a hypothesis.

## 6.2 NSA Surveillance

Our results bear directly on the NSA's technical capabilities, against individuals both within and external to the United States. For non-U.S. individuals, our data indicates that the agency could have access to a majority of a person's browsing history (71.3% of visits in Ireland and 61.1% of visits in Japan), without ever collecting data outside the United States.[11] Furthermore, we show that the agency can link a non-trivial portion of this traffic through cookies.

Nearly a quarter of U.S. page visits are visible outside the U.S. through third parties and referers. While we show that linking is infeasible, this does not imply that U.S. users browsing domestic sites are safe from NSA surveillance. Passive tracking techniques, or a static IP address, could cause the user's traffic to be just as vulnerable.

## 6.3 Mitigation by users

Figure 3 shows that users can minimize their exposure to surveillance through cookies, but can not eliminate it all together. Since the identifiers of advertising networks play a key part in the transitive linking of page visits, ad filtering lists (e.g. Ghostery) are currently the most effective solution for users. However even after blocking these lists, ISP level identifiers like Verizon's UIDH would cause the vulnerability to persist [26]. Firefox's built-in cookie blocking can also be effective in reducing the level of traffic clustering, though even the most restrictive option leaves nearly a third of a user's traffic vulnerable to clustering.

Users have very little control over identity leakage outside of stopping the use of services. Forcing HTTPS connections after login can help (i.e. using HTTPS Everywhere), but as we show in our measurements, two-thirds of sites still do not support HTTPS after this step is taken. Users can also be careful to not reuse usernames between sites as this could provide an additional identifier to link page visits.

## 6.4 Mitigation by trackers

Trackers can prevent a passive eavesdropper from piggybacking on their unique identifiers if they are only transmitted over HTTPS. Some trackers have already deployed HTTPS to avoid mixed content warnings when embedding in HTTPS pages. There are also subtle issues, such session

[11] Alternatively, the agency could rely on collection points just outside the U.S. That approach would fall under Executive Order 12333, which affords more latitude than Section 702 of the FISA Amendments Act.

tickets used for TLS session resumption, which can be used by an eavesdropper to link multiple HTTPS connections to the same browser instance similar to a cookie.

Unfortunately, a large fraction of trackers would need to deploy such mitigation strategies for them to make a dent in the adversary's overall chances. As we showed in Section 5.4 with our HTTPS Everywhere measurements, the feasibility of traffic clustering only drops by 23% when the amount requests occurring over HTTPS more than doubles.

## 6.5 Limitations

A couple of important limitations of the attack must be pointed out. First, using the Tor browser bundle likely defeats it. "Cross-origin identifier unlinkability" is a first-order design goal of the Tor browser, which is achieved through a variety of mechanisms such as double-keying cookies by first-party and third-party [44]. In other words, the same tracker on different websites will see different cookies. However, our measurements on identifier leakage on popular websites apply to Tor browser usage as well. Preventing such leakage is not a Tor browser design goal.

Simply disabling third-party cookies will also deter the attack, but it is not clear if it will completely stop it. There are a variety of stateful tracking mechanisms in addition to cookies [14, 23, 51, 37, 6], although most are not as prevalent on the web as cookies are.

We also mention two limitations of our study. First, while a significant fraction of popular sites transmit identities of logged-in users in the clear, we have not actually measured how frequently typical users are logged in to the sites that do so. Anecdotal evidence suggests that this number must be high, but experiments on actual user sessions are required for an accurate estimation of vulnerability.

Second, we use commercial geolocation data with an additional custom metric to determine if requests enter the United States for users in Japan and Ireland, and to determine if requests leave the United States for users within the U.S. Even with this additional check, two scenarios can occur: requests outside the U.S. can be marked as entering the U.S. due to an incorrect geolocation and high network latency, and requests inside the U.S. can be marked as staying in the U.S. if they are incorrectly geolocated as being further away than the actual location of the destination (which may still be external to the U.S.).

## 6.6 Other applications of our methodology

Stated in general terms, we study an adversary with a given set of technical capabilities, network positions, and policy restrictions, and ask, for a given user browsing model, how much of her activity is vulnerable. Our general algorithm in Section 4 can be easily adapted to study a variety of questions that fit this framework. For example, we might assume an adversary who can can compel certificate creation and uses this to spoof some third parties on secure websites to launch active attacks. Here the relevant measurement would be the prevalence of active third-party content on websites visited by typical users and the geographic distribution of third parties serving such content.

# 7. CONCLUSION

While much has been said from a legal, ethical and policy standpoint about the recent revelations of NSA tracking, many interesting *technical* questions deserve to be considered. In this paper we studied what can be inferred from the surveillance of web traffic and established that utilizing third-party tracking cookies enables an adversary to attribute traffic to users much more effectively than methods such as considering IP address alone. We hope that these findings will inform the policy debate on tracking, raise user awareness of subtle, inferential privacy breaches, and lead to the development of better defenses and greater adoption of existing ones.

# 8. ACKNOWLEDGMENTS

We would like to thank Jennifer Rexford, Doug Madory, Harlan Yu, and our anonymous reviewers for their insightful comments, as well as Andrew Clement and Colin McCann for their willingness to share data related to this study.

# 9. REFERENCES

[1] ShareMeNot: Protecting against tracking from third-party social media buttons while still allowing you to use them. `https://sharemenot.cs.washington.edu`.

[2] TrackingObserver: A Browser-Based Web Tracking Detection Platform. `http://trackingobserver.cs.washington.edu`.

[3] Executive Order 12333–United States intelligence activities. `http://www.archives.gov/federal-register/codification/executive-order/12333.html`, 1981.

[4] NSA 'planned to discredit radicals over web-porn use'. `http://www.bbc.co.uk/news/technology-25118156`, November 2013.

[5] 'Tor Stinks' presentation - read the full document. `http://www.theguardian.com/world/interactive/2013/oct/04/tor-stinks-nsa-presentation-document`, October 2013.

[6] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz. The Web never forgets: Persistent tracking mechanisms in the wild. In *Conference on Computer and Communications Security (CCS)*. ACM, 2014.

[7] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens, and B. Preneel. FPDetective: dusting the web for fingerprinters. In *Conference on Computer and Communications Security (CCS)*. ACM, 2013.

[8] A. Arnbak and S. Goldberg. Loopholes for circumventing the constitution: Warrantless bulk surveillance on americans by collecting network traffic abroad, 2014.

[9] M. Ayenson, D. J. Wambach, A. Soltani, N. Good, and C. J. Hoofnagle. Flash cookies and privacy II: Now with HTML5 and ETag respawning. *World Wide Web Internet And Web Information Systems*, 2011.

[10] M. Balakrishnan, I. Mohomed, and V. Ramasubramanian. Where's that phone?: geolocating IP addresses on 3G networks. In *Internet Measurement Conference (IMC)*. ACM, 2009.

[11] R. Balebako, P. Leon, R. Shay, B. Ur, Y. Wang, and L. Cranor. Measuring the Effectiveness of Privacy Tools for Limiting Behavioral Advertising. In *Web 2.0 Security & Privacy (W2SP)*. IEEE, 2012.

[12] J. Ball. NSA stores metadata of millions of web users for up to a year, secret files show. `http://www.theguardian.com/world/2013/sep/30/nsa-americans-metadata-year-documents`, 2013.

[13] P. E. Black. Ratcliff/Obershelp pattern recognition. http://xlinux.nist.gov/dads/HTML/ratcliffObershelp.html, December 2004.

[14] E. Bursztein. Tracking users that block cookies with a HTTP redirect. http://www.elie.net/blog/security/tracking-users-that-block-cookies-with-a-http-redirect, 2011.

[15] S. Chen, R. Wang, X. Wang, and K. Zhang. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Security and Privacy (S&P)*. IEEE, 2010.

[16] A. Clement. IXmaps–Tracking your personal data through the NSA's warrantless wiretapping sites. In *International Symposium on Technology and Society (ISTAS)*. IEEE, 2013.

[17] B. Elgin and V. Silver. The Surveillance Market and Its Victims. http://www.bloomberg.com/data-visualization/wired-for-repression/, 2011.

[18] S. Englehardt, C. Eubank, P. Zimmerman, D. Reisman, and A. Narayanan. Web Privacy Measurement: Scientific principles, engineering platform, and new results. Manuscript, 2014.

[19] R. Gallagher. Operation Socialist: The Inside Story of How British Spies Hacked BelgiumâĂŹs Largest Telco. https://firstlook.org/theintercept/2014/12/13/belgacom-hack-gchq-inside-story/, 2014.

[20] Ghostery. Are we private yet? http://www.areweprivateyet.com/.

[21] S. Gorman and J. Valentino-Devries. New Details Show Broader NSA Surveillance Reach. http://online.wsj.com/news/articles/SB10001424127887324108204579022874091732470, 2013.

[22] G. Greenwald and S. Ackerman. How the NSA is still harvesting your online data. http://www.theguardian.com/world/2013/jun/27/nsa-online-metadata-collection, 2013.

[23] M. Hastak and M. J. Culnan. Persistent and unblockable cookies using HTTP headers. http://www.nikcub.com/posts/persistant-and-unblockable-cookies-using-http-headers, 2011.

[24] D. Herrmann, R. Wendolsky, and H. Federrath. Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naive-Bayes Classifier. In *Workshop on Cloud Computing Security (CCSW)*. ACM, 2009.

[25] A. Hintz. Fingerprinting Websites Using Traffic Analysis. In *Privacy Enhancing Technologies*. Springer, 2003.

[26] J. Hoffman-Andrews. Verizon Injecting Perma-Cookies to Track Mobile Customers, Bypassing Privacy Controls. https://www.eff.org/deeplinks/2014/11/verizon-x-uidh, 2014.

[27] B. Krishnamurthy, K. Naryshkin, and C. Wills. Privacy leakage vs. Protection measures: the growing disconnect. In *Web 2.0 Security & Privacy (W2SP)*. IEEE, 2011.

[28] B. Krishnamurthy and C. Wills. Privacy diffusion on the Web: a longitudinal perspective. In *International Conference on World Wide Web (WWW)*. ACM, 2009.

[29] B. Krishnamurthy and C. E. Wills. On the Leakage of Personally Identifiable Information Via Online Social Networks. In *Workshop on Online Social Networks (WOSN)*. ACM, 2009.

[30] B. Krishnamurthy and C. E. Wills. Privacy leakage in mobile online social networks. In *Conference on Online Social Networks (COSN)*. USENIX, 2010.

[31] M. Lee. Secret "BADASS" Intelligence Program Spied on Smartphones. https://firstlook.org/theintercept/2015/01/26/secret-badass-spy-program/, 2015.

[32] B. Liu, A. Sheth, U. Weinsberg, J. Chandrashekar, and R. Govindan. AdReveal: Improving Transparency Into Online Targeted Advertising. In *Workshop on Hot Topics in Networks (HotNets)*. ACM, 2013.

[33] D. Madory, C. Cook, and K. Miao. Who Are the Anycasters? In *Proceedings of NANOG59*, 10 2013.

[34] D. Malandrino, A. Petta, V. Scarano, L. Serra, and R. Spinelli. Privacy awareness about information leakage: Who knows what about me? In *Workshop on Privacy in the Electronic Society (WPES)*. ACM, 2013.

[35] J. Mayer. Tracking the Trackers: Self-Help Tools. https://cyberlaw.stanford.edu/blog/2011/09/tracking-trackers-self-help-tools, September 2011.

[36] J. Mayer and E. W. Felten. The Web is Flat. http://webpolicy.org/2013/10/30/the-web-is-flat/, 2013.

[37] J. R. Mayer and J. C. Mitchell. Third-party web tracking: Policy and technology. In *Security and Privacy (S&P)*. IEEE, 2012.

[38] A. M. McDonald and L. F. Cranor. Survey of the use of Adobe Flash local shared objects to respawn HTTP cookies. *ISJLP*, 7:639, 2011.

[39] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of Tor. In *Security and Privacy (S&P)*. IEEE, 2005.

[40] S. J. Murdoch and P. Zieliński. Sampled Traffic Analysis by Internet-Exchange-Level Adversaries. In *Privacy Enhancing Technologies*. Springer, 2007.

[41] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Security and Privacy (S&P)*. IEEE, 2013.

[42] L. Olejnik, T. Minh-Dung, C. Castelluccia, et al. Selling Off Privacy at Auction. 2013.

[43] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel. Website Fingerprinting in Onion Routing Based Anonymization Networks. In *Workshop on Privacy in the Electronic Society (WPES)*. ACM, 2011.

[44] M. Perry, E. Clark, and S. Murdoch. The design and implementation of the Tor browser [DRAFT]. https://www.torproject.org/projects/torbrowser/design, November 2014.

[45] F. Roesner, T. Kohno, and D. Wetherall. Detecting and Defending Against Third-Party Tracking on the Web. In *Networked Systems Design and Implementation (NDSI)*. USENIX, 2012.

[46] 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.

[47] A. Soltani, A. Peterson, and B. Gellman. NSA uses Google cookies to pinpoint targets for hacking. http://www.washingtonpost.com/blogs/the-

switch/wp/2013/12/10/nsa-uses-google-cookies-to-pinpoint-targets-for-hacking, December 2013.

[48] D. X. Song, D. Wagner, and X. Tian. Timing Analysis of Keystrokes and Timing Attacks on SSH. In *Security Symposium*. USENIX, 2001.

[49] A. M. White, A. R. Matthews, K. Z. Snow, and F. Monrose. Phonotactic reconstruction of encrypted VoIP conversations: Hookt on fon-iks. In *Security and Privacy (S&P)*. IEEE, 2011.

[50] T.-F. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi. Host fingerprinting and tracking on the web: Privacy and security implications. In *Network and Distributed System Security Symposium (NDSS)*. IEEE, 2012.

[51] M. Zalewski. Rapid history extraction through non-destructive cache timing (v8). http://lcamtuf.coredump.cx/cachetime/.