Johnny Still Can't Opt-out: Assessing the IAB CCPA Compliance Framework

Muhammad Abu Bakar Aziz
Northeastern University
Boston, MA, USA
aziz.muh@northeastern.edu

Christo Wilson
Northeastern University
Boston, MA, USA
cbw@ccs.neu.edu

ABSTRACT

The privacy laws and regulations that govern the collection, sharing, and selling of online data are changing. In the U. S., California adopted the California Consumer Privacy Act (CCPA) and the California Privacy Rights Act (CPRA), and twelve other U. S. states have adopted similar laws. Industry has responded by developing technical standards for collecting and disseminating consent information, such as the IAB CCPA Compliance Framework. While publishers are adopting this framework and the IAB is extending it to cover privacy laws in other U. S. states, recent work has observed that opt-out signals are not being honored under the framework.

In this study, we take a deep dive into the IAB CCPA Compliance Framework to measure end-to-end flows of consent information and better understand why opt-out signals are not being honored. Using data crawled from top websites under four different experimental conditions, we examine overall adoption of the framework, the flow of consent information from publishers to third parties and between third parties, and finally the reach of opt-out signals. Our results uncover numerous issues with the adoption and implementation of the framework that prevent users' consent choices from being honored by third parties.

KEYWORDS

California Consumer Privacy Act, Internet Advertising Bureau U. S. Privacy Framework, Global Privacy Control

1 INTRODUCTION

The privacy laws and regulations that govern the collection, sharing, and selling of online data are changing. Starting in Europe, the original governing framework of notice and consent [19] has given way to the ePrivacy Directive [52] and the General Data Protection Regulation (GDPR) [22]. In the U. S., California adopted the California Consumer Privacy Act (CCPA) [9, 10] and then strengthened it by adopting the California Privacy Rights Act (CPRA) [11]. Twelve other U. S. states have adopted similar laws [20].[1]

These laws are multifaceted, but for our purposes their most salient features are their consent requirements. The GDPR requires that people opt-in to the collection and processing of their data for specific purposes, such as online advertising [18]. The U. S. state laws instead grant people the right to opt-out of data selling and sharing under specific circumstances. Either way, to comply with these laws, online *advertising and analytics (A&A) companies* must now be consent-aware, as their ability to collect, share, and sell data depends in part on individual consent choices.

A&A companies have responded to these laws by developing technical standards for collecting and disseminating consent information. In Europe, the Transparency & Consent Framework (TCF)—developed by the Interactive Advertising Bureau (IAB) Europe [23]—is a standard that specifies the parties that are authorized to collect consent from people (*Consent Management Platforms* or *CMPs*), the encoded data format used to store consent choices (the *consent string*), and the JavaScript API that *publishers*,[2] CMPs, and third-party A&A companies use to communicate. Although the TCF is a voluntary standard, in practice it has seen significant adoption [16, 40], in part because it has survived the scrutiny of European regulators [17].

After the passage of the CCPA, the IAB CCPA Compliance Framework [32] (which we abbreviate as the CCPA Framework) was adopted as the analogue of the TCF. Like its European predecessor, the CCPA Framework defines an encoded format for storing consent information and a JavaScript API for accessing this information. Publishers are adopting this framework [12, 62] and the IAB is extending it to cover privacy laws around the world [25].

However, investigations have uncovered numerous problems with the implementation and usage of the TCF in practice. These problems include, but are not limited to, tracking that occurs before people make a consent choice [40] and websites that fail to honor opt-out requests [15, 16, 47, 51, 55]. Recent work found that, in California, users' choices to opt-out were not always accurately captured by the CCPA Framework [12], and that opting-out did not have a substantive impact on data sharing and selling practice [39]. These findings suggest that similar technical problems may be impacting consent frameworks in Europe and the U. S.

In this study, we investigate whether the CCPA Framework is able to faithfully convey Californians' consent choices from publishers to third parties. We focus on the CCPA Framework because it is the industry standard for CCPA/CPRA compliance, even though it is a voluntary standard like the TCF. Our study goes beyond prior work on CCPA compliance by examining end-to-end flows of consent information, in contrast to just examining the flow from the web browser to the publisher [12]. Examining how consent information is or is not flowing to third parties is crucial for understanding why Californians' consent choices are being ignored by advertisers [39]. We investigate the following research questions:

[1] As of March 2024 these states are: Colorado, Connecticut, Delaware, Indiana, Iowa, Montana, New Jersey, Oregon, Tennessee, Texas, Utah, and Virginia.

[2] This is the ad industry term for websites that are ad-supported.

Proceedings on Privacy Enhancing Technologies 2024(4), 349–363

https://doi.org/10.56553/popets-2024-0120

- **RQ1**: Are top publishers—those who are likely covered by the CCPA and CPRA—adopting the CCPA Framework?
- **RQ2**: To what extent and how is the U. S. Privacy String [31] (which we abbreviate as USP String) passing from publishers to third parties?
- **RQ3**: To what extent is the USP String being passed among third parties?
- **RQ4**: Are users' decisions to opt-out (via Global Privacy Control [24] and the USP API) being passed to third parties, and if so, are third parties altering their practices accordingly?

To collect data for our study, we performed four crawls—between April 2023 and February 2024 using IP addresses in California—of the top 10 K websites in the Tranco ranking [36] of popular domains. During two crawls our crawler did nothing to opt-out of data selling and sharing, while in two it communicated a choice to opt-out. We instrumented our crawler to detect the presence of the U. S. Privacy User Signal API (the standard for reading the USP String, which we abbreviate as USP API), as well as record reads and writes of the USP API by first- and third-party JavaScript, writes of first-party cookies (where the USP String is often stored), all HTTP requests and responses, and the full *inclusion tree* of each webpage (i.e., the graph representation of all causal loading relationships between resources in the page) [3, 6].

Our analysis reveals a number of potential issues with the CCPA Framework. Some of these issues are rooted in low adoption: by publishers that have not adopted the USP API, and by A&A companies that do not read the USP String or pass it consistently to other third parties via HTTP requests. Other issues are rooted in ambiguity in the standard itself, such as third parties that bypass the official USP API when reading the USP String, store copies of the USP String in non-standard cookies, or pass the USP String in non-standard URL parameters. Ultimately, our opt-out tests demonstrate that A&A companies, in general, do not appear to be respecting Californians' consent choices.

Our findings, like those from prior work, are important for at least two reasons. First, it is up to regulators in Europe and the U. S. to enforce online privacy laws, and regulators must be informed about potential problems with industry consent standards. Second, the IAB is in the process of deploying a new technical standard, the Global Privacy Platform [25], that is meant to harmonize consent collection and dissemination on the global Internet. This standard draws on the existing technologies of the TCF and CCPA Framework, so it is likely to suffer from the same issues as its predecessors.

The outline of our study is as follows. We begin in § 2 by presenting an overview of online privacy laws, associated technical standards, and prior work. In § 3 we introduce our data collection methodology. In § 4 we present the analysis of our data and discuss outcomes in § 5.

2 BACKGROUND AND RELATED WORK

We begin by discussing online privacy laws in Europe, their associated technical standards for communicating consent, and studies that have identified issues in Europe. Next, we discuss the same topics in the U. S. Finally, we discuss universal opt-out signals, which play a role in our study.

2.1 The GDPR and the TCF

The GDPR [22]—implemented in 2018—fundamentally altered the online privacy landscape in Europe. Along with the ePrivacy Directive [52], which was implemented in 2002, the GDPR stipulates that website publishers must have a legally valid basis for processing non-essential personal data—which includes, for example, unique identifiers, browsing history, and geolocation data commonly used to target online ads—with one such basis being user consent. The need for user consent has led to proliferation of Consent Management Platforms (CMPs) [28] and consent banners (or dialogs) on websites that collect this required information from users [16].

To help publishers and A&A companies comply with the GDPR, the IAB Europe introduced the TCF in 2018. In this framework, publishers include JavaScript code into their website from a CMP. The CMP's code is responsible for at least three functions: (1) displaying a cookie banner to collect consent if it has not already been collected for a given user, (2) storing the correctly encoded consent string, and (3) providing mechanisms for A&A companies to read and share the consent string. At a minimum, the code provided by CMPs must implement JavaScript APIs for reading the consent string; they may also provide access to a cookie containing the consent string. Additionally, the TCF states that the consent string can be shared via HTTP requests using any one of three URL parameters: `gdpr`, `gdpr_consent`, and `gdpr_pd`. The most recent version of the TCF is 2.2 [23].

Numerous studies have documented problems with cookie banners in Europe. This includes the use of *dark patterns* that manipulate users into making sub-optimal privacy choices [27, 41, 58] and non-functional banners that fail to accurately record users' choices [40]. Although issues with consent banners are critical, they are not the focus of our study.

Similarly, several studies have found that opting-out under the auspices of the ePrivacy Directive or GDPR does not produce substantive privacy benefits for users. Studies have observed, at most, a small reduction in tracking cookies after opting-out [15, 16, 47, 55] and a negligible reduction in resource inclusions from A&A companies [51]. Matte et al. [40] identified suspected GDPR violations on publishers that had adopted the TCF, while Smith et al. [50] observed that 73% of publishers with the TCF continued to share data even after people declined to consent by claiming they had a "legitimate interest" exemption. For our study we leverage measurement techniques from this body of work to assess similar issues in the U. S. context, e.g., whether A&A companies are obeying technical standards and whether opting-out reduces tracking.

2.2 CCPA/CPRA and CCPA Framework

The California Consumer Privacy Act (CCPA) [9, 10]—which went into effect in 2020—was the first comprehensive online privacy law enacted in the U. S. Like the GDPR, the CCPA codifies the notion of consent for sharing and selling of personal data; unlike the GDPR, however, the CCPA enacts an opt-out regime, meaning A&A companies may collect and monetize users' personal data unless users affirmatively opt out.[3] The CCPA mandates that publishers include a hyperlink with the text "Do Not Sell Or Share My Personal

[3] See CA Civ. Code §1798.120(a).

Information" on their homepage[4] through which people may opt-out of data selling and sharing [30, 60] and some publishers have also adopted consent banners [42].

In response to the ratification of the CCPA, the IAB introduced the CCPA Framework in 2019 [32]. The general functionality of the CCPA Framework is similar to that of the TCF (see the previous section). Specifically, the CCPA Framework requires that a JavaScript method called `__uspapi()` be instantiated in the first-party context. This method must support a `getUSPData` command that returns a `uspData` object containing the USP String [31]. This method can be called directly by third parties present in the first-party context, or indirectly using the JavaScript `postMessage` DOM API to communicate with a special `__uspapiLocator` iframe. The CCPA Framework recommends that the USP String be stored in a first-party cookie named `usprivacy` and that it be shared using a URL parameter with the name `us_privacy`. The CCPA Framework does not specify whether and how consent choices should be presented to users; user interface design is left to parties that provide implementations of the Framework, such as CMPs.

The CCPA Framework defines the format of the USP String using a four character long encoding [31]:

- '1', indicating the version of the USP String.
- 'N', 'Y', or '-', indicating whether the publisher provided notice to the user of their right to opt-out.
- 'N', 'Y', or '-', indicating whether the user has opted out of data selling and sharing.
- 'N', 'Y', or '-', indicating whether the publisher has signed the IAB's Limited Service Provider Agreement.

In the last three cases, a dash indicates that the given component was not applicable to the given user. For example, a USP String value of `1YNN` indicates that the CCPA applies to a given user, they were presented with notice, and they have not opted out; `1YYN` indicates the user has opted out; and `1---` indicates that the CCPA does not apply to this user. In this study, we focus on the third character of the USP String, which conveys opt-out choices.

Recent studies have identified problems with publishers' CCPA compliance that are similar to what has been observed in Europe with GDPR compliance. O'Connor et al. [42] found dark patterns that hinder users' ability to activate their opt-out rights in "Do Not Sell" links and consent banners. Liu et al. [39] performed opt-out experiments in California and found that they had no statistically significant impact on digital advertising practices. Charatan and Birrell [12] found websites where the value of the USP String did not accurately reflect opt-out signals from the user's web browser, which helps to partially explain the findings from Liu et al. [39]. In this study, our goal is to analyze the flow of consent information to and between third parties (not just from the browser to the publisher), focusing on the CCPA Framework and the USP String, to help explain why opting out does not appear to be effective even in cases where websites implement the CCPA Framework.

2.3 Determining CCPA/CPRA Applicability

A key difference between the GDPR and the CCPA is their applicability. The GDPR applies to all publishers that reside or target people in Europe. The CCPA, in contrast, applies to publishers that reside or target people in California, sell or share users' data, and meet at least one prerequisite of a three-part test: (1) earn at least $25 M USD per year, (2) sell the personal data of at least 50 K Californians, or (3) derive at least 50% of their revenue from the sale of Californians' data [60].[5] The CPRA [11]—which went into effect in 2023—increased the second prerequisite of the three-part test to 100 K Californians.

There is no definitive method for an outsider to determine whether the CCPA or CPRA applies to a given website because the information necessary for making this determination (i.e., revenue and counts of unique visitors from California) is not public. A website that includes a "Do Not Sell" link or an implementation of the USP API may have self-determined that the CCPA or CPRA applies to them, but these signals are not entirely reliable. For example, prior work has found unpopular websites—i.e., ones that are unlikely to meet the revenue or unique visitor requirements of the laws—that include "Do Not Sell" links [60]. Prior work has also observed websites that implement the USP API but do not include a "Do Not Sell" link, which is an ambiguous configuration—should this be interpreted as an incomplete attempt to comply with the law [62]?

Van Nortwick and Wilson [60] estimated that the CCPA and CPRA may apply to top 10 K and 5 K websites in the Tranco ranking [36], respectively, because these websites meet the second requirement of the three-part eligibility test. They made this determination by leveraging estimates of unique visitors to websites that they acquired from a marketing firm. Based on these results we focus our study on the top 10 K websites in the Tranco ranking.

2.4 Global Privacy Control

A unique feature of the CCPA is that it incorporates the concept of "user-enabled global privacy controls" that publishers and A&A companies must comply with [8].[6] The Global Privacy Control (GPC) standard is recognized as one such control under the CCPA/CPRA [45, 62]. GPC is functionally identical to the Do Not Track (DNT) standard: in both cases, when the setting is enabled in a browser, the browser communicates the user's opt-out decision in all HTTP requests and makes the opt-out status available via a DOM API. For GPC, these respective mechanisms are the `Sec-GPC: 1` HTTP header and the `navigator.globalPrivacyControl == True` DOM property [46]. DNT is not recognized as global privacy control under CCPA/CPRA.

The California Attorney General successfully settled a complaint against Sephora for not complying with GPC [45], which demonstrates the importance of this opt-out technique. However, recent work has found that publishers and CMPs in California were not complying with GPC opt-out requests [12, 62]. In this study, we extend this prior work by examining how GPC and USP String opt-out requests propagate through publishers and third parties, to better understand whether or not they are being honored.

3 METHODS

In this section we describe the web crawls that we performed to collect data for our study. Table 1 presents an overview of our four crawls, including the experimental conditions used in each

[4] See CA Civ. Code §1798.135(a)(1).

[5] See CA Civ. Code §1798.140(c)(1).

[6] See CA Department of Justice regulations, 11 CA ADC §999.315(c).

Name	Date	Injected USP API?	USP String Value	GPC	Used in §	Used in Figures	Used in Tables	Crawled %	A&A %
Crawl 1	Apr.2023	No	–	Off	4.1, 4.2.2, 4.3.2, 4.3.3, 4.3.4, 4.4	1, 2, 6, 7, 8, 9, 10	2, 4, 5, 6	99.67	67.99
Crawl 2	Feb.2024	Yes	1YYN	Off	4.2.1, 4.3.1, 4.4	3, 4, 5, 9, 10	3	99.91	69.00
Crawl 3	Dec.2023	No	–	On	4.4	9, 10		99.62	63.56
Crawl 4	Jan.2024	No	–	Off	4.4	9, 10		97.54	64.78

Table 1: Crawls we performed to gather data. GPC was enabled in one crawl, and in one crawl we injected the USP API into each webpage and set the value it returned for the USP String to 1YYN (i.e., opted-out of data selling and sharing). All crawls covered 10 K websites and were performed using IP addresses in California. We present the percentage of domains in each crawl that successfully returned at least one HTTP(S) response and the percentage that embedded at least one resource from an A&A third party.

and references to where we use each dataset. Each crawl served a specific purpose:

- **Crawl 1** provides baseline measurements about the adoption of the USP API and the default values of the USP String from the perspective of a user in California in 2023. Other than being instrumented to record data, this crawler was unmodified. It did not interact with consent dialogs or signal any attempt to opt-out of data collection or sale.
- In **Crawl 2** and **Crawl 3**, we configured our crawler to opt-out of data selling and sharing using the USP API or GPC, respectively. We use these datasets to examine whether the choice to opt-out is being communicated to and respected by A&A companies.
- **Crawl 4** repeated **Crawl 1** in early 2024. This crawl provides longitudinal perspective on whether our findings regarding adoption of the USP API generalize over time.

We describe each crawl in detail below.

3.1 Website Selection

To gather data for this study, we chose to crawl the top 10 K domains from the Tranco list [36].[7] We focus on the top 10 K domains because Van Nortwick and Wilson [60] found that the CCPA and CPRA were unlikely to apply to websites that fell below this level of popularity since they did not receive enough unique visitors from California to meet the laws' eligibility criteria (see § 2.3). That said, the CCPA and CPRA may not apply to all domains in this list—e.g., domains owned by non-profit organizations—and thus we refrain from asserting whether specific websites are in compliance with the CCPA or CPRA (see § 3.5). Rather, the goal of our study is to assess the overall adoption of the CCPA Framework and flows of consent information, a goal for which it is sufficient for us to cover popular websites. Our selection criteria is more conservative than that used by Charatan and Birrell [12], who considered the top 25 K domains in the Tranco list.

3.2 Baseline Crawler Configuration

Each of our four crawls used the same baseline configuration. **Crawl 1** and **Crawl 4** used this baseline configuration with no modifications. The remaining crawls included additional modifications as described in § 3.3 and § 3.4.

[7] We use the Tranco list dated April 17th, 2023, with ID GZ7NK.

3.2.1 Overview. We used custom scripts, written in Python and JavaScript, to drive and instrument an instance of Chrome[8] using the Chrome DevTools Protocol [13]. We left Chrome at its default settings, except during crawls where we varied HTTP headers, as described below. All crawls were conducted using virtual machines from Amazon Web Services with IP addresses in California.

During each crawl of the Tranco top 10 K, our crawler visited each domain one-by-one. For each domain, we programmed the crawler to load the domain's homepage,[9] scroll to the bottom of the page, then sleep for 25 seconds. Further, we programmed our crawler to select nine internal hyperlinks at random from the homepage and crawl them using the same load, scroll, and sleep approach. We crawled homepages and subpages because prior work has shown that they can behave differently [2, 56].

We assessed the impact of anti-crawler countermeasures on our crawler by manually revisiting 200 randomly selected websites, weighted by Tranco rank, from **Crawl 3** and **Crawl 4**, using the same IP addresses as the crawler used. We received CAPTCHA challenges on two of the websites that prevented them from loading normally. Thus, we estimate that around 1% of websites in our sample were impacted by anti-crawler countermeasures.

3.2.2 Inclusion Trees and Chains. Our crawler recorded detailed information during each visit to a webpage, including all HTTP request and response headers and all cookies that were set. Furthermore, our crawler recorded the resource inclusion tree for each webpage [3, 6]. The inclusion tree is a data structure that captures the causal loading relationships between all resources that comprise a webpage. The root of the inclusion tree for a given webpage is the base HTML document, the nodes are JavaScript or other documents (e.g., in iframes), and the directed edges indicate that the parent resource caused the child resource to be loaded. Each directed edge is associated with an HTTP request and response.

As in prior work, we use inclusion information to understand the relationships and flows of data between first and third parties [4, 6, 7, 60]. We decompose the inclusion tree for each webpage into *inclusion chains*, where each chain corresponds to a unique path

[8] We used Chrome version 108.0.5359.124 with user-agent string "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36".

[9] Our crawler attempted to visit each domain in Tranco first using HTTPS and then HTTP. If the domain was unreachable, the crawler would prepend 'www.' to the domain and try again to visit using HTTPS and then HTTP.

from root to leaf in the given tree [5]. Furthermore, given the focus of our study, we isolated *A&A chains* that correspond to the serving of an ad or a tracker. We label a given inclusion chain as an A&A chain if (1) there was at least one HTTP request in the chain that matched a rule in the EasyList or EasyPrivacy block lists,[10] or (2) the chain terminated in the loading of a 1×1 tracking pixel [21]. We use these A&A chains in § 4 to analyze the sources and destinations of HTTP requests that included the USP String, i.e., to understand how this consent signal is being passed from one party to another.

3.2.3 Detecting the USP API. The CCPA Framework mandates that compliant publishers implement the USP API within their application context [31]. To understand which publishers support this API and what default value the USP String had been set to, we programmed our crawler to inject a content script into the first-party execution context of each crawled webpage 25 seconds after loading the page. Our script first attempted to detect the presence of the `__uspapi()` method. If it was present, then our script called the method and recorded the resulting USP String. This is a similar approach to the one used by Matte et al. [40] to detect the presence of cookie banners in webpages that implemented TCF version 1.1.

We manually validated our crawler's ability to detect the USP API and USP String value. To assess false positives we randomly selected 50 websites, weighted by Tranco rank, from **Crawl 4** where our crawler detected the USP API and revisited them manually in Chrome using an IP address in California. Our crawler successfully detected the USP API on 49 websites, yielding a false positive rate of 2%. Furthermore, the value of the USP String recorded by our crawler matched our manual observation of the value (in the Chrome developer tools) of the USP String in 96% of cases. Collectively, our results are similar to the 3% false positive rate of the USP String crawler used by Charatan and Birrell [12].

To assess false negatives we randomly selected 50 websites, weighted by Tranco rank, from **Crawl 4** where our crawler did not detect the USP API and did detect at least one embedded resource from an A&A company. We manually revisited these websites and found zero false negatives.

3.2.4 Tracing USP Cookie Writes. The CCPA Framework recommends that publishers store the USP String in a first-party cookie named `usprivacy`. Since this is only a suggestion, it is possible that publishers or third parties may also store the USP String in other first-party cookies.

To understand which parties were writing first-party cookies, we instrumented our crawler to record all accesses to the DOM `cookie.set` method. We used code from DuckDuckGo's Tracker Radar Collector (TRC) [54] that allowed us to set non-invasive breakpoints on the DOM cookie methods and record the JavaScript stack traces, which include the parameter values passed to the method and the origin of the script calling the method. TRC has been successfully used by prior work to study security and privacy-relevant behaviors of websites [49, 53]. The stack traces collected by our crawler enabled us to attribute DOM-based cookie writes to JavaScript from specific parties, which we analyze in § 4.[11]

[10] https://easylist.to

[11] Our instrumentation also tracks DOM-based reads of first-party cookies, but this data is hard to interpret because the API returns all first-party cookies. Thus, we cannot tell which specific cookie or cookies are of interest to each reader.

3.3 USP API Instrumentation and Opt-out

The instrumentation in our baseline crawler allows us to determine which publishers implement the `__uspapi()` method on their website. However, the baseline instrumentation does not enable us to determine which parties (first or third) were instantiating the API, or which parties, if any, were invoking the API to read the USP String. Furthermore, because our baseline crawler did not interact with consent banners and "Do Not Sell" links, it cannot communicate opt-out choices via the USP API by publishers (and CMPs) that are using the CCPA Framework.

To answer these questions and rectify these shortcomings, we developed our own implementation of the USP API. We used the `Page.addScriptToEvaluateOnNewDocument` functionality of the Chrome DevTools Protocol to inject our USP API implementation into webpages before they loaded [44], and we set our implementation as non-configurable and non-writeable [34] so that our implementation could not be preempted or overwritten by other parties. Any attempt to overwrite our USP API implementation resulted in a JavaScript exception error that was recorded by our crawler. The captured exceptions included a stack trace through which we could identify the origin of the JavaScript attempting to instantiate the USP API. Similarly, we designed our USP API implementation to throw a non-fatal exception whenever its methods were invoked, thus enabling us to determine the origin of the JavaScript that was calling the method in question.

During **Crawl 2**, we injected our custom USP API with the `__uspapi()` method configured to always return the USP String value `1YYN` (which encodes a choice to completely opt-out of data selling and sharing). Thus, we directly leveraged the CCPA Framework to convey the user's intent to opt-out without needing to interact with consent banners.

We used the dataset produced in **Crawl 2** to identify non-standard cookies and URL parameters that stored a USP String. We searched all cookie values and URL parameter values for our chosen USP String value and manually validated their contents. We then manually examined 20 randomly selected URLs that contained the same name–value pair from **Crawl 1** to check whether they followed the USP String format. We observed some URL parameters that appeared to store values other than the USP String, such as unique identifiers. While these parameters raise privacy concerns, they are clearly not designed to be a mechanism for conveying user consent. Therefore, we exclude these parameters from our analysis. Matte et al. [40] also observed URL parameters containing consent information that did not adhere to the TCF 1.1 specifications. We present the cookie and URL parameter names that contained the USP String in Table 2 and Table 3.

We validated that our USP API injection approach was successful by testing it. Our crawler called the `__uspapi()` method after injecting the USP API into each crawled webpage and recorded the USP String value. Out of 10 K crawled websites, we observed our chosen USP String value in 9,982 (99.8%) websites. We manually reviewed the 18 domains where our injection failed and found that most were either a non-public page from a Content Delivery Network or were failing to load. This demonstrates that our injection was successful and the opted-out USP String was available to be read via the `__uspapi()` method on most webpages during

Crawl 2. Furthermore, we observed our chosen USP String value in the `us_privacy` parameter in URLs, which confirms that at least some parties read and transmitted it.

3.4 GPC Opt-out

During **Crawl 3**, we enabled GPC in our crawler by adding the `Sec-GPC: 1` header to all HTTP requests and setting the `navigator.globalPrivacyControl` property to true. We manually validated our crawler's ability to detect the USP API when GPC was enabled. To assess false positives we randomly selected 50 websites, weighted by Tranco rank, from **Crawl 3** where our crawler had detected the USP API. We manually revisited these websites using an IP address in California and enabled GPC in our browser. Our crawler successfully detected the USP API on 47 websites, yielding a false positive rate of 6%. Additionally, we confirmed that the GPC functionality of our crawler worked by having it visit the official GPC validation website [24].

3.5 Research Ethics

We followed standard ethical research practices when collecting data for our study. To ensure that our crawler did not overburden web servers, we designed it to only collect 10 pages per domain spaced out over five minutes. Our crawler did not click on or otherwise interact with advertisements or consent banners. Our crawler did not honor *robots.txt* files, as doing so could enable adversarial websites to hide bad behavior—e.g., failing to honor users' opt-out choices—by forbidding crawlers. The fidelity of our study hinges on the ability to browse websites as normal users do, and thus we designed our browser accordingly.

Our approach for injecting the USP API into websites causes no harm to them. This is the same technique used by browser extensions to add or remove functionality from websites, e.g., to block ads or translate the language of text. Our crawler's script injection occurs on the client side and only affects the webpages loaded in our browser; it has no impact on the website's server or other users of the website.

Throughout our analysis and discussion, we are careful not to make legal determinations about the compliance of particular publishers or A&A companies with privacy laws. Making these kinds of determinations requires careful, individualized analysis and legal expertise that is beyond the scope of our work.

4 RESULTS

In this section, we present the results of our analysis. First, we present the publisher-level analyses to understand the overall adoption of the CCPA Framework. Second, we examine communication between publishers and third-party JavaScript in terms of reads and writes to the interfaces of the USP API. Third, we trace communication of the USP String between third parties. Finally, we investigate the impact of opting-out on the behavior of third parties.

4.1 Adoption by Publishers

We begin by assessing the adoption of the CCPA Framework by publishers in the Tranco top 10 K. For this analysis we rely on data from our baseline **Crawl 1** from California. We examine adoption of the official USP API as well as the use of official and unofficial cookies for storing the USP String.

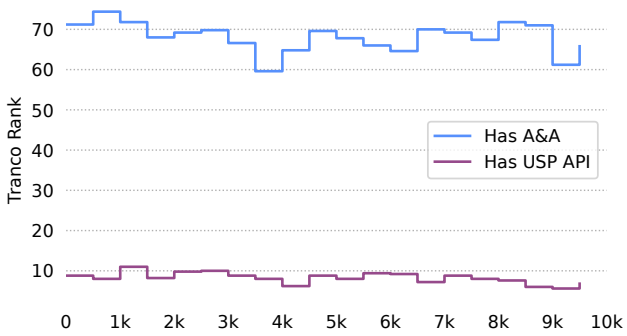

Figure 1: Percentage of websites in Crawl 1 that adopt the USP API and/or embed A&A resources, bucketed into groups of 500 by Tranco rank.

4.1.1 USP API. In **Crawl 1** we detected the presence of the USP API on 821 (8.2%) publishers.[12] This agrees with findings from Charatan and Birrell [12], who observed that 8.3% of websites in the Tranco top 25 K adopted the USP API. Similarly, Matte et al. [40] found that 6.2% of 22,949 websites they crawled in 2019 had adopted the TCF.

Out of 821 publishers, 438 were within the top 5 K websites, i.e., the range where the CPRA is likely to be applicable [60]. If we only consider the top 5 K websites, this pushes the adoption rate of the USP API up to 8.7%. While these adoption rates appear to be low, it may be the case that some publishers in our sample do not need to adopt the USP API because the CCPA/CPRA do not apply to them, or because they are not selling or sharing data to or with third parties. If we further restrict our consideration to the websites among the top 5 K that include A&A resources (3,425 websites), then the USP API adoption rate is 12.8%.

To better contextualize the adoption rate that we observe, we plot Figure 1, which presents the fraction of websites in **Crawl 1** that (1) adopt the USP API and/or (2) include objects from well-known A&A domains (see § 3.2.2), bucketed in groups of 500 by Tranco rank. We observe that adoption of the USP API and A&A resources is relatively invariant to rank, likely because our crawls focus on large, sophisticated, well-resourced publishers. However, it is troubling that adoption of the USP API is low given that 6,799 (68.0%) of these websites include A&A resources.

Our findings differ from those reported by Hils et al. [29], who found that TCF adoption among the Tranco top 5 K websites was above 12%. TCF adoption may outpace CCPA Framework adoption because it applies to more websites (see § 2.3).

Next, we examine the categories of publishers that incorporate the USP API. We use the domain-to-category mapping developed by Fortiguard because Vallina et al. [59] found that it had the greatest

[12] We consider a publisher to support the USP API if it was accessible on at least one of their webpages. Across **Crawl 1**, **Crawl 3**, and **Crawl 4**, the USP API adoption rate varied between 8.2–8.5%.

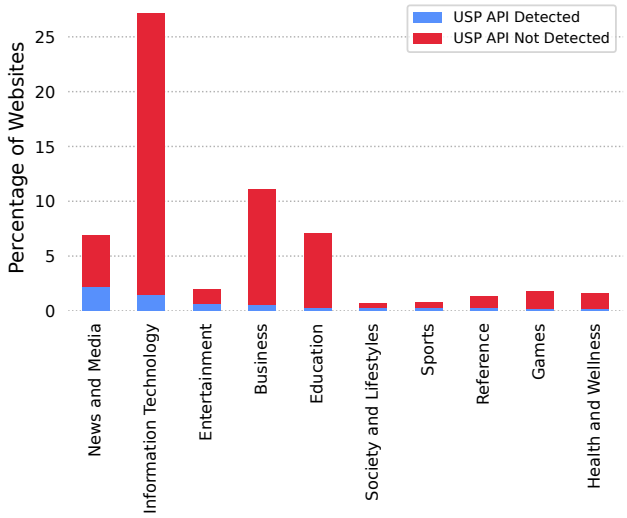

Figure 2: Percentage of websites in Crawl 1 that did and did not adopt the USP API, grouped by Fortiguard categories. For brevity, we focus on the top ten categories among websites that adopted the USP API.

Cookie Name	# of Websites
usprivacy	321
ntv_as_us_privacy	54
us_privacy	13
ccpa	5

Table 2: Standard and non-standard cookie names that stored a USP String value in Crawl 1 (n = 10 K).

coverage and highest accuracy versus other mappings. The Fortiguard API returned categories for 9897 (98.9%) of the websites in our corpus, spanning 75 different categories. It returned categories for all 821(100%) of the websites that had adopted the USP API. Figure 2 shows the percentage of websites in **Crawl 1** that did and did not adopt the USP API, grouped by Fortiguard categories and sorted by USP API adoption. For brevity, we focus on the ten largest categories, which cover 78.7% of the 821 websites in this analysis. We observe that USP API adoption is more prevalent on publishers in the News and Media, Information Technology, and Entertainment categories. Our results are similar to those from Hils et al. [29], who found that TCF adoption was highest among publishers in the News & Entertainment category. These results are intuitive, given that these publishers are most likely to monetize their content via online ads. We also observe, however, that in no category did the majority of publishers adopt the USP API.

4.1.2 USP Cookies. In our data from **Crawl 1** we detect the USP String value being stored in a cookie on 358 (3.6%) publishers out of 10 K, of which 321 are using the recommended `usprivacy` cookie. Table 2 presents the cookie names that we observe storing a USP String value, along with their usage frequencies. We find that the cookie name `usprivacy` is used most frequently, but also that there are a significant number of non-standard names used in practice. Storing the USP String in a non-standard cookie or in a location other than a cookie (e.g., Local Storage) is potentially problematic, as third parties that bypass the USP API and read the USP String directly from non-standard storage may not find the expected value.

Furthermore, we observed 35 publishers that had two different cookies each storing a USP String value. For example, we observed a `usprivacy` and a `ntv_as_us_privacy` cookie on 25 websites. This is also potentially problematic, as the USP String values stored in multiple cookies may not match, meaning that third parties may read a USP String value that does not accurately reflect a user's consent choice.

Next, we examine the extent to which publishers conform with the full breadth of the CCPA Framework by looking at the co-occurrence of the USP API and the `usprivacy` cookie. To simplify our analysis, we focus solely on homepages from **Crawl 1** because we observe variability in USP API and `usprivacy` cookie adoption across the webpages from a given publisher.[13] Out of 10 K publishers, we observed that 267 homepages included both the USP API and `usprivacy` cookie, 15 had only the `usprivacy` cookie, and 438 had only the USP API. It is not surprising that the USP API is more prevalent overall since the CCPA Framework mandates the use of the API, whereas it only recommends the `usprivacy` cookie for storing the USP String [31]. That said, it is concerning to observe so much heterogeneity in the implementation of the USP API and its underlying storage mechanism given that the CCPA Framework is intended to be a universal mechanism for conveying user consent decisions in California. These results about the inconsistency of USP API and `usprivacy` cookie co-adoption are similar to those observed by Zimmeck et al. [62], who found that among 64 websites with the USP String, three only stored it in the `usprivacy` cookie, 25 only made it available via the USP API, and 36 had both.

4.2 First to Third Party Interfaces

At this point, we have examined the adoption and implementation of the CCPA Framework by publishers. Next, we examine how publishers and third parties interact by examining the instantiation of the USP API, as well as reads to the USP API and USP Cookies.

4.2.1 USP API. We begin by examining instantiation of and reads to the USP API. Recall that we can only examine these events when we inject our own implementation of the API (see § 3.3). Thus, for the following analysis, we rely on the data from **Crawl 2**.

In total we observe attempts to instantiate the USP API on 234 publishers,[14] which is less than the 821 publishers that we observed adopting the USP API under normal conditions.[15] We suspect that some first and third parties may use JavaScript reflection to determine whether the USP API exists before attempting to instantiate their own implementation of the API. Thus, our results for USP API instantiation are a lower bound and may miss third parties who instantiate the API conservatively.

[13] For example, publishers often omit third-party scripts from specific webpages, such as their privacy policies, that our crawler may randomly visit.
[14] Recall that attempts to overwrite our USP API implementation result in JavaScript errors that are recorded by our crawler, see § 3.3.
[15] Out of these 234 publishers, 183 had the USP API in **Crawl 1**.

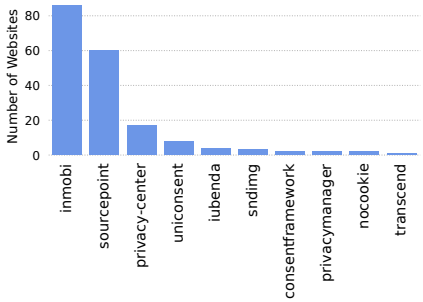

Figure 3: Top ten third-party domains trying to install the USP API during Crawl 2.

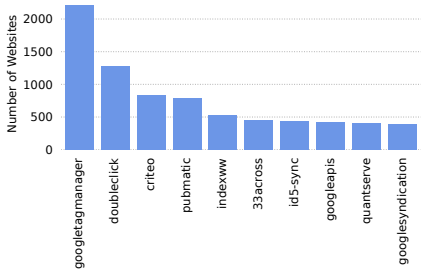

Figure 4: Top ten third-party domains reading the USP String from the USP API during Crawl 2.

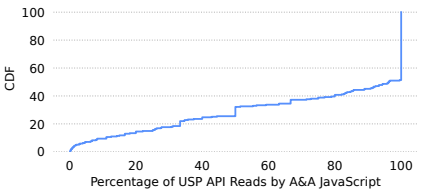

Figure 5: Consistency of reading the USP API by A&A JavaScript during Crawl 2. Each point in the distribution is the percentage of times when JavaScript from a given A&A domain read the USP API when it was loaded.

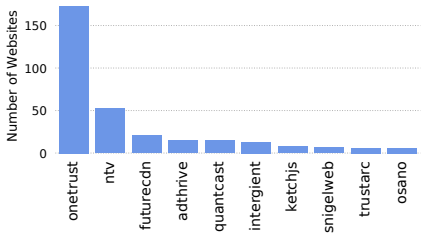

Figure 6: Top ten third-party domains writing to a USP Cookie during Crawl 1.

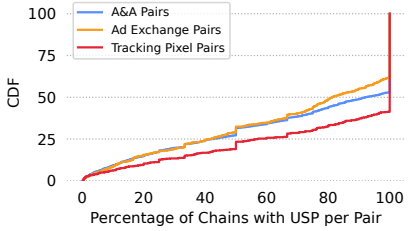

Figure 7: Consistency of sharing the USP String between pairs during Crawl 1. Each point in the distribution is the percentage of chains containing a unique initiator–receiver pair where the former sent an HTTP request containing the USP String to the latter.

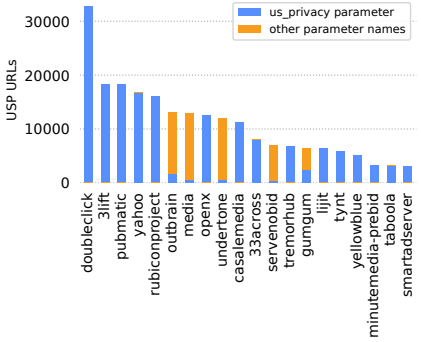

Figure 8: Top 20 third-party domains that receive the USP String during Crawl 1, stratified by the URL parameter name.

Figure 3 presents the top ten third parties attempting to instantiate the USP API. Unsurprisingly, we find that the list is dominated by CMPs—e.g., Sourcepoint, UniConsent, Consent Framework by Sirdata, and iubenda—with Inmobi (which acquired Quantcast in 2023) [33] being most prevalent (instantiating the USP API on 86 publishers). Measurements conducted by Hils et al. in 2020 and 2021 found that OneTrust, Quantcast, TrustArc, Source Point, and LiveRamp were popular CMPs for implementing the TCF in Europe [28, 29]. Comparing our findings to those from Hils et al. highlights that the landscape of consent management services is changing over time, possibly because new markets are emerging as online privacy laws change.

Regarding reads from the USP API, Figure 4 presents the top 15 third parties that read from the `__uspapi()` method. Unlike instantiation, the list of readers is dominated by A&A companies, with Google being most prevalent (Google Tag Manager reads the `__uspapi()` method on 2,218 publishers; Doubleclick reads on 1,275), followed by Criteo (reads on 829 publishers). Interestingly, although we only observe the USP API on 821 publishers in **Crawl 1**, we find that third parties read the USP API on 3590 publishers in **Crawl 2** (i.e., when our crawler injects the USP API and makes it available on all 10 K publishers). This highlights a disparity between the number of publishers that are actively attempting to convey consent information to third parties versus the frequency at which third parties are attempting to receive this information from publishers.

Finally, we examine how consistent A&A third parties are at reading the USP API. First, we locate all JavaScript resources that originate from A&A domains and were embedded in publishers during **Crawl 2**. In total we identify 2,631 A&A domains that have at least one script embedded on at least one publisher. Out of these, we identify 257 A&A domains whose scripts called the USP API on at

least one publisher. Finally, for each of these 257 A&A domains, we calculate the percentage of times its script read the USP API when it was loaded and plot the results in Figure 5.[16] We find that 48.6% of these A&A domains read the USP API consistently, when they load, while the remaining A&A scripts read inconsistently. There are many possible reasons for these inconsistencies, including race conditions (i.e., the A&A script executes before the USP API has been instantiated) and different versions of scripts from a given origin (e.g., older versions of scripts may not be aware of the USP API). Note that 90% of A&A domains with scripts never read the USP API in our dataset, so the vast majority are consistently ignoring the USP API.

4.2.2 USP Cookies. Next, we examine writes to the four cookies that we observe storing a USP String (see Table 2). Recall that our crawler recorded all invocations of the DOM cookie API (see § 3.2.4). As such, for this analysis we rely on the data gathered during **Crawl 1**.

Figure 6 presents the top ten third parties that write the USP cookies. OneTrust, a popular CMP, is the most frequent writer, doing so on on 173 publishers. Other frequent writers include Nativo (a native advertising company), AdThrive (an influencer marketing company), Quantcast (in their role as a CMP), and Ketch (another CMP). We observe five websites on which two different parties write to the `usprivacy` cookie, which is problematic because it may lead to a classic data race condition.

Overall, we do not observe a large number of websites where USP cookies are being written. This is potentially explained by the mechanics of the CCPA/CPRA and the design of our crawler. Because these are opt-out laws, publishers and CMPs do not necessarily need to record a USP String by default. Further, because our crawler does not attempt to use "Do Not Sell" links or consent banners to opt-out, our crawler never triggers JavaScript that would write a USP String to storage. It is also possible that the USP String is being stored in locations other than cookies.

4.3 Consent Flow Among Third Parties

In the previous section we observed that various third parties are reading the USP String directly from the USP API (and possibly also from first-party cookies). However, third parties may also receive the USP String indirectly from other third parties. Indeed, it is well documented that A&A companies routinely share data, e.g., through the distribution of bid requests in ad exchanges or through cookie syncing [1, 6, 7, 43]. Thus, in this section, we examine whether third parties are transmitting the USP String to each other through URL parameters in HTTP requests.

4.3.1 USP String in URL Parameters. Table 3 presents the URL parameter names that we observed transmitting a USP String value, along with their usage frequencies. As with cookie names, we find that the URL parameter name recommended by the CCPA Framework, `us_privacy`, is used most frequently, but also that non-standard names are widely used. Matte et al. [40] also observed non-standard parameter names being used to transmit TCF consent strings, but they did not quantify the frequency of these practices.

[16] For example, if JavaScript from Criteo was loaded on 1,003 websites and read the USP API on 830 of them, then its read percentage is 82.8%.

Parameter Name	# of Websites
us_privacy	3360
ccpa	358
usp_consent	280
uspString	223
sst.us_privacy	167
uspConsent	143
ccpa_consent	92
AV_CCPA	89
usp	83
usprivacy	74
_fw_us_privacy	68
D9v.us_privacy	62
cnsnt	45
ccpaconsent	37
usp_string	33

Table 3: Top 15 most frequent standard and non-standard URL parameter names that stored a USP String value. We present the number of publishers on which we observed each name using data from Crawl 2 (n = 10 K).

In the data collected during our **Crawl 2** we observed that 421,497 HTTP requests contained a URL with a USP parameter, of which 354,416 (84.1%) contained our chosen USP String value (`1YYN`). This implies that the majority of parties are reading the USP String from the USP API (or receiving a copy from another party who read the USP API) and then including the correct USP String value in their HTTP requests.[17]

We hypothesize several reasons why the remaining 15.9% of HTTP requests did not contain our chosen value of the USP String. First, the originators of these HTTP requests may be retrieving the USP String from some storage location not covered by our instrumentation. Second, the originators of these requests may be disregarding the USP API entirely and passing a default USP String value in all HTTP requests.

4.3.2 USP String in Inclusion Chains. Next, we shift from studying individual URLs to inclusion chains, each of which captures end-to-end resource loading over sequences of URLs (see § 3.2.2). For the sake of ecological validity, we switch to analyzing the inclusion chains collected during **Crawl 1**. In this dataset, we observe 319,269 HTTP requests that contain the USP String. Interestingly, we observe HTTP requests that transmit the USP String from 1,033 publishers, yet we find only 821 publishers adopting the USP API, which suggests that some third parties include a default USP String in their HTTP requests even when the USP API is not present on the publisher.

Overall, this dataset contains 3,102,021 A&A inclusion chains, but 60% of these chains are rooted in publishers that do not adopt the USP API, thus it is not reasonable to expect A&A companies to transmit the USP String in their HTTP requests. There are 1,214,540 A&A inclusion chains rooted in the 821 publishers that implement

[17] This data may include false positives if a given initiator always transmits a default USP String value of `1YYN`.

Initiator	Uniq. Receivers
doubleclick	128
rubiconproject	109
cloudfront	59
alternet	53
sltrib	51
confiant-integrations	48
casalemedia	32
aniview	30
taboola	29
googleapis	28
pubmatic	27
makeuseof	24
thoughtcatalog	24
heavy	24
wtop	23

Table 4: Top 15 domains that sent the USP String, sorted by unique receivers during Crawl 1 (n = 10 K).

Receiver	Uniq. Initiators
rubiconproject	270
bidswitch	245
3lift	169
pubmatic	168
yahoo	158
openx	155
doubleclick	144
a-mo	121
adsrvr	107
casalemedia	102
33across	97
yieldmo	93
media	78
1rx	75
criteo	70

Table 5: Top 15 domains that received the USP String, sorted by unique initiators during Crawl 1 (n = 10 K).

Initiator	Receiver	# of Chains
doubleclick	doubleclick	15144
outbrain	outbrain	10959
pubmatic	pubmatic	8388
indexww	casalemedia	8349
primis	primis	7529
googlesyndication	doubleclick	7477
rubiconproject	rubiconproject	6385
taboola	tremorhub	5319
servenobid	yellowblue	4537
33across	tynt	4410
yahoosandbox	yahoo	3429
aniview	aniview	3392
servenobid	minutemedia-prebid	3165
rubiconproject	3lift	2963
casalemedia	yahoo	2921

Table 6: Top 15 initiator/receiver pairs sorted by total chains during Crawl 1 (n = 10 K).

the USP API, of which 218,541 (17.9%) contained at least one HTTP request transmitting the USP String. 171,866 (78.6%) of these A&A chains included exactly one HTTP request transmitting the USP String, despite the median A&A chain length being five. These findings are problematic, as they reveal that most HTTP requests between A&A companies do not contain users' consent information at all, and that USP String sharing during complex, multi-step operations (e.g., serving programmatic ads) is limited.

4.3.3 Initiators and Receivers. Next, we study the sources (*initiators*) and destinations (*receivers*) of HTTP requests that include at least one of the URL parameters listed in Table 3 (and the value of this parameter is not null). We continue to rely on inclusion chains from **Crawl 1** for this analysis.

Table 4 presents the top 15 third parties that we observe initiating HTTP requests that contain the USP String, sorted by the unique number of third parties that receive these HTTP requests. Many of the most prolific distributors of the USP String are ad exchanges (e.g., Doubleclick, Rubicon Project, Casale Media, Aniview, and Taboola), which makes intuitive sense: the JavaScript served by these companies sends HTTP requests to many other third parties, depending on the outcome of RTB auctions. Table 5 presents the top 15 third parties that we observed receiving HTTP requests that contain the USP String, sorted by the unique number of initiators. Again, we observe many ad exchanges on the list, likely because many third parties match cookies with them.

Table 6 shows the top 15 initiator/receiver pairs that we observe exchanging the USP String, sorted by the total number of chains in which the pairs appear. The most striking aspect of this list is that nine of the top 15 pairs are within the same company (for example, indexww and casalemedia are both owned by Index Exchange).

4.3.4 Pairwise Analysis. Although it is a positive sign that we observe many major A&A companies sending and receiving the USP String, our analysis thus far does not indicate whether each company is doing so consistently. For the CCPA Framework to succeed, consent must be communicated consistently to each A&A company and in a consistent format.

Figure 7 examines the former requirement: consistency of USP String sharing between A&A companies. First, we isolate all the inclusion chains from **Crawl 1** that are rooted on the 821 publishers that adopted the USP API, i.e., cases where a USP String was available. Second, we identify all pairs of initiators and receivers in this subset of inclusion chains where at least one member of the pair is an A&A third party. We refer to these as *A&A pairs*. Finally, for each A&A pair, we calculate the percentage of A&A chains in which the former sent an HTTP request to the latter that contains a USP String (again, restricted to inclusion chains rooted in the 821 publishers that adopted the USP API) and present the empirical cumulative distribution over all A&A pairs. Note that we exclude all A&A pairs that never shared the USP String in any HTTP requests (which removes 90% of all A&A pairs).

Ideally, the distribution over all A&A pairs in Figure 7 should be close to 100%, indicating that when pairs communicate they consistently share the USP String. Unfortunately, we observe that only 47.1% of A&A pairs share the USP String in all chains in which they appear. The remaining pairs exhibit varying levels of inconsistency, which is potentially problematic as the receivers may depend on the USP String data from the initiators to honor users' opt-out choices.

In addition to plotting the distribution over A&A pairs in Figure 7, we also plot the distribution when we restrict to pairs where (1) at least one member is a major ad exchange,[18] or (2) the receiver is downloading a tracking pixel. We focus on these two types of pairings because they have particular importance in the advertising ecosystem. Ad exchanges disseminate bid requests containing personal information to hundreds of third parties as part of RTB auctions [7], so it is critical that they receive and disseminate users' preferences regarding data collection. Tracking pixels are ubiquitous on the Web [21] and they have specific requirements with respect to opt-out signals: because they are images, not JavaScript, they cannot access the `__uspapi()` method or the `navigator.globalPrivacyControl` property, so the only way they can receive and act on opt-out signals is via HTTP requests.[19]

As we show in Figure 7, the dissemination of the USP String is still inconsistent even when we restrict our investigation to pairs

[18] We focus on the 30 most popular ad exchanges based on frequency data from `ads.txt` files gathered by Bashir et al. [5].

[19] Alternatively, tracking pixels can simply not be downloaded at all.

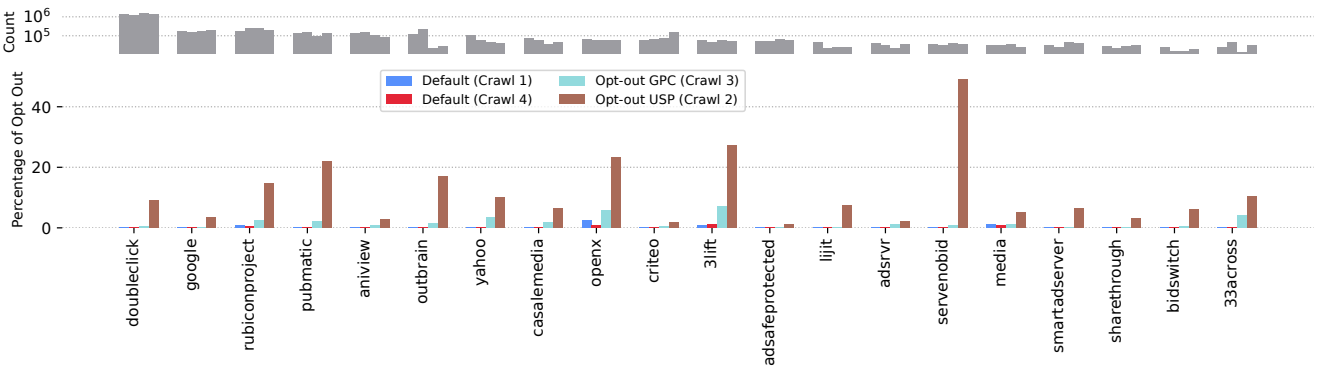

Figure 9: Count of HTTP requests sent to third parties (top) and percentage of those requests containing a USP String set to opt-out (bottom) across four crawls. Crawl 2 and Crawl 3 included opt-out signals via the USP API and GPC, respectively. The third parties along the x-axis are sorted by the count of requests in Crawl 1. We focus on the top 20 domains where at least 1% of their incoming HTTP requests contain a USP String set to opt-out in Crawl 2.

that include an ad exchange or a tracking pixel. In these cases, 38.3% and 58.6% of pairs, respectively, share the USP String in all chains in which they appear. Note that we exclude 75.2% of ad exchange pairs and 93.3% of tracking pixels pairs that never shared the USP String in any HTTP requests (see Figure 7).

To examine whether the USP String is being shared between A&A pairs via a consistent URL parameter name, Figure 8 shows the count of HTTP requests received by the top receivers that included a USP String stratified by the URL parameter name: either the recommended `us_privacy` parameter or any other name (see Table 3). Overall, 231,648 (79.8%) of these URLs use the `us_privacy` parameter, and we see that most of the top receivers are sent the USP String in this name consistently. Five receivers are sent the USP String using a non-standard name the majority of the time, and in these cases we also observe a minority of requests using the recommended `us_privacy` parameter. These latter cases are potentially problematic: in practice, each A&A company must define an API that specifies how it expects to receive consent information via HTTP. If HTTP requests attempt to pass consent signals using an unsupported format (i.e., an incorrect URL parameter), the consent signal may be lost and the user's wishes will not be honored.

4.4 Impact of Opting-out

In the previous sections we identified a number of potential problems with the current adoption and implementation of the CCPA Framework. In this section, we take the next logical step and examine whether users' opt-out decisions are being respected. We examine two opt-out mechanisms—USP API and GPC—and rely on data from two crawls—one with our injected USP API that sets USP String to opt-out of the selling and sharing of data (**Crawl 2**; see § 3.3), and one with GPC enabled (**Crawl 3**, see § 3.4). Our crawler provides these opt-out signals on all publishers, so if there are effects we should observe them on all crawled websites, not just the 821 publishers that adopt the USP API.

To establish baseline expectations, we first examine whether the USP String encodes an opt-out decision–the third character in USP String set to 'Y' (see § 2.2)- under normal conditions using data from **Crawl 1**. In **Crawl 1**, 24 out of 821 (2.92%) publishers with the USP API set the USP String to opt-out despite the fact that our crawler did not interact with these websites in any way. It is unclear why these publishers have configured their website to opt their visitors out of data selling and sharing by default (or even whether they are aware of this, as it may be a misconfiguration). In **Crawl 3**, when GPC was enabled, 380 out of 825 (46.1%) publishers with the USP API set the USP String to opt-out.[20] Zimmeck et al. [62] found that 12.5% of the 64 websites with the USP API that they studied correctly set the USP String to opt-out when they received a GPC signal, while Charatan and Birrell [12] observed ~38% of the ~2 K websites with the USP API in their study correctly set the USP String to opt-out. Collectively, these results suggest that some publishers are reacting to the GPC signal by correctly setting the USP String to opt-out—thus helping to convey the users' opt-out intent to third parties—but more than half are not. This is potentially problematic, as third parties may rely on the publisher to observe and redistribute the GPC signal via the USP API.

Next, we contrast how the A&A third parties behave in the absence and presence of opt-out signals. We compare data gathered in four crawls: our baseline crawls (**Crawl 1**, **Crawl 4**), our crawl when the USP API returned the USP String set to opt-out (**Crawl 2**), and our crawl with GPC enabled (**Crawl 3**). We measure behavior in two respects: (1) how many total HTTP requests are made to A&A third parties and (2) how many HTTP requests to A&A third parties include a USP String set to opt-out? The first question is relevant because publishers and third parties may respond to opt-out signals by retrieving fewer resources (e.g., tracking pixels and cookie syncing) from A&A companies. The second question is relevant because A&A third parties who are contacted should be informed about users' decisions to opt-out if they are to comply with them.

[20] Additionally, 48 publishers set the `gpcEnabled` field to true in the `__uspapi()` method's output, which helps confirm that they did react to our crawler's GPC signal.

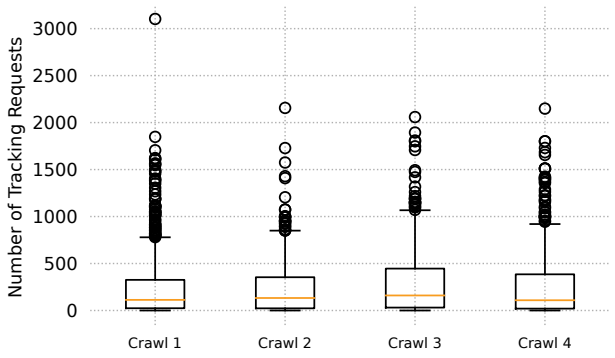

Figure 10: Count of tracking pixels per publisher across all four of our crawls. Crawl 2 and Crawl 3 included opt-out signals via the USP String and GPC, respectively.

Figure 9 shows the count of HTTP requests and the percentage of those requests that contain a USP String set to opt-out for the top 20 A&A domains (sorted by total requests) across our four crawls. We omit domains that did not receive at least 1% of HTTP requests with a USP String set to opt-out in **Crawl 2**. We observe some effects in the data from **Crawl 3**: some A&A domains receive HTTP requests containing the USP String, and only Outbrain exhibits a noticeable drop in total HTTP requests. All of these A&A companies directly received the GPC signal from our crawler, so it is possible that they may still react to the signal correctly (e.g., by not permanently storing data about the user), but we still interpret these findings as being potentially problematic.

We observe increased dissemination of the USP String set to opt-out when we compare the data from **Crawl 2** to the data from **Crawl 3**: four of the A&A domains shown in Figure 9 receive a USP String set to opt-out in >20% of incoming HTTP requests. Based on these observations, it is clear that distributing opt-out signals in California via the CCPA Framework has more impact than distributing them using GPC alone. These improvements are relatively modest, however, given that the vast majority of HTTP requests to A&A companies continue to occur and do not include a USP String set to opt-out. For example, Google (Tag Manager, Syndication, Analytics), and Facebook receive <1% of requests containing a USP String set to opt-out.

Finally, we performed a targeted analysis looking specifically at the impact of opt-outs on tracking pixels.[21] Figure 10 presents a box plot of the number of tracking pixels per publisher across our four crawls. To make this analysis comparable across crawls, we focus on the 793 publishers that (1) adopted the USP API and (2) were present in all four crawls.

Figure 10 shows that tracking pixels are widely adopted across all four crawls, even when opt-out signals are present in **Crawl 2** and **Crawl 3**. While we do find statistically significant differences in the distribution of tracking pixels per publishers when comparing **Crawl 1** and **Crawl 3** (two-sided t-test, $p < 0.05$), the effect sizes are small (Cohen's $d < 0.1$). We do not observe statistically significant differences when we compare **Crawl 2** or **Crawl 4** to **Crawl 1**. Based on these results we conclude that GPC and a USP String set to opt-out do not significantly reduce the number of tracking pixels embedded in publishers, and that any variations can be attributed to natural variability in tracking pixel inclusion over time.

[21] Defined as A&A chains that end with the download of a 1×1 image.

5 DISCUSSION

Returning to our research questions, we reach a number of troubling conclusions from our study. We find that overall adoption of the CCPA Framework is low (8.2%) among the top 10 K publishers on the Web (**RQ1**), even though 68.0% of these publishers embed A&A resources. Other studies have also found low overall adoption rates of consent technologies in California [12, 39, 60, 62].

With respect to the interface between publishers and third parties (**RQ2**), we observe many issues that may result in incorrect USP String values being read and disseminated or correct values being erroneously overwritten. These include: USP String storage in non-standard cookies and multiple cookies at once (§ 4.1.2), third parties that never read the USP API or do so inconsistently (§ 4.2.1), and cases where multiple parties write the `usprivacy` cookie (§ 4.2.2).

Regarding the flow of USP String consent information among third parties (**RQ3**), while we observe over 1.2 M A&A inclusion chains that could have conveyed a USP String, only 17.9% did so. Further, the vast majority of chains only contained a single instance of the USP String being conveyed—often by a company sending it back to themselves (see Table 6)—despite each chain typically including multiple communications between multiple A&A companies. Further, we observed that 52.9% of A&A pairs that we observed communicating do not pass the USP String to each other consistently. Complicating matters further, we observed that some third parties have not adopted the recommend `us_privacy` parameter standard for passing the USP String, and that this creates room for ambiguity where these parties are being passed the USP String in potentially incorrect URL parameters.

Lastly, our experiments demonstrate how the failures we document surrounding the USP API culminate in a system that does not effectively disseminate users' opt-out decisions or alter the behavior of A&A companies (**RQ4**). Like prior work [12, 62], we observe that publishers and CMPs systematically fail to convert GPC signals into the USP String set to opt-out via USP API, and thus any third party relying on the USP API to convey accurate consent information will be misinformed. We do observe more dissemination of the opt-out signal in the HTTP requests via our injected USP API but these improvements are modest. Most importantly, we find very little substantive alteration in behavior in the presence of opt-out signals: downloads of tracking pixels are unaffected, and only a few third parties experience a large reduction in resource requests.

5.1 Is the CCPA Framework Working?

Overall, we conclude that the CCPA Framework is not working, for a variety of definitions of "working". The standard does not appear to be well adopted by publishers or by A&A companies, and among the implementations that we observe there are potentially problematic technical and usage practices that hinder the clear and unambiguous communication of consent information (e.g., the use of non-standard cookie and parameter names). Now, on one

hand, the CCPA Framework is not a mandatory standard, so it is perhaps not surprising to find less-than-universal adoption. On the other hand, however, it is the direct analog to the TCF and it is supported by the IAB, which is the primary trade group for digital advertisers. The CCPA has been in effect since January 2020, thus it is troubling to observe that the U. S. digital advertising industry has not coalesced around a single technical specification for eliciting, storing, and disseminating consent information.

Unfortunately, our findings demonstrate that history is repeating itself in California, since we observe similar classes of technical problems that have been documented in Europe with the TCF [16, 40]. Our study builds on and extends studies documenting deficiencies in CCPA/CPRA compliance in California [60, 62]. Our study helps to explain the findings of Liu et al. [39], by demonstrating how intentional and unintentional failure modes prevent A&A companies from receiving and acting upon users' consent data.

5.2 Could the GPP Work?

The IAB is deprecating the CCPA Framework and replacing it with the Global Privacy Platform (GPP) [25], which is meant to provide a uniform encoding format and API for consent signals around the world (e.g., in Europe, Canada, and the U. S.). Given the issues that we and others have documented with the TCF and the CCPA Framework, we ask: could the GPP address these problems?

To some extent, it could try. If national regulators required adherence to the GPP standard, it could drive increased adoption by publishers and A&A companies in the U. S. Further, the GPP standard could attempt to mitigate technical failings by mandating that CMPs and third-party JavaScript obey best practices, such as: mandating use of official APIs to read the consent string, forbidding redundant copies of the consent string in storage, and only using official names for passing the consent string. That said, because the GPP, like the TCF and CCPA Framework, is instantiated using embedded JavaScript, it cannot prevent deviations from the standard by imposing technical restrictions. For example, there is no technical measure that can prevent race conditions between multiple third parties writing the consent string to storage.

For these reasons, we do not believe standards developed by the IAB, or any other advertising industry consortium, offer the appropriate tools for collecting, managing, and disseminating users' consent information. In our opinion, we believe that the correct location to implement consent functionality is in the browser itself. Only the browser can (1) make consistent, immutable DOM APIs available in all first- and third-party contexts for querying consent information and (2) guarantee that consent information is communicated in all HTTP requests. Further, the user-agent is best positioned to present a clear (i.e., free of dark patterns [26, 41, 58]) and consistent user interface for making consent choices. Thus, we support ongoing efforts to enforce compliance with GPC [45] and develop a robust user interface to this functionality [14, 37, 61]. As we have shown, there are multiple points of failure when transferring consent information from the user to A&A companies via the CCPA Framework. We advocate that A&A companies should be required to act on GPC opt-out signals that they receive directly from browsers, rather than requiring the needless and complicated involvement of publishers or IAB frameworks.

5.3 Is the USP API a Declaration of Applicability?

Determining whether a given website falls within the applicability criteria of the CCPA or CPRA is not a trivial task. Prior work has leveraged the presence of the USP API on a website as a marker of self-professed applicability by the publisher [62]. We caution that this approach to determining CCPA/CPRA applicability is not sound: because the USP API is implemented in JavaScript, any party can install it in a given context, including without the publisher's consent. Web developers routinely do not understand script inclusions in their websites [35, 48, 57] and this is especially true with respect to A&A scripts because they are highly dynamic and resource loading is non-deterministic (e.g., due to cookie matching).

5.4 Limitations

Our study has several limitations. First, there may be CMPs and third parties that encode consent information in formats that are not compliant with the CCPA Framework that we cannot detect. Thus, our results should be interpreted as lower bounds on the propagation of consent signals. Similarly, because our crawler injected its own USP API implementation rather than interacting with publishers' "Do Not Sell" links and consent banners, we may not capture privacy-preserving functionality that is only activated by interacting directly with CMPs. That said, we did not observe a significant reduction in the sharing and selling of data when our crawler opted-out via GPC, which is a legally binding mechanism that CMPs should recognize and respond to. Californians who enable GPC in their browser should not have to click on "Do Not Sell" links or consent banners to activate their privacy rights.

Second, we cannot fully assess third parties' compliance with opt-out signals because we cannot observe how data is handled server-side (e.g., whether it is used to develop cross-context advertising profiles or sold to other parties). It is possible that some of the third parties we observed receiving opted-out USP String values did obey them even if they continued to load resources. That said, we observed very limited propagation of these opt-out signals, so we expect the impact of hidden compliance to be minimal, and other studies provided corroborative evidence that compliance with opt-out signals in the U. S. is low [12, 39, 62].

Finally, our study does not account for third parties operating under the CCPA's service provider exception [38]. This exception allows them to continue storing users' data, which they can sell or share even when users have opted out.

5.5 Data Release

The data and code to produce figures and analysis are available at https://github.com/abubakaraziz/Assessing-IAB-CCPA-Framework.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments. This research was supported in part by NSF grant IIS-1910064. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

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 Conference on Computer and Communications Security*.

[2] Waqar Aqeel, Balakrishnan Chandrasekaran, Anja Feldmann, and Bruce M Maggs. 2020. On Landing and Internal Web Pages: The Strange Case of Jekyll and Hyde in Web Performance Measurement. In *Proceedings of the ACM Internet Measurement Conference*. 680–695.

[3] Sajjad Arshad, Amin Kharraz, and William Robertson. 2017. Include Me Out: In-browser Detection of Malicious Third-party Content Inclusions. In *Proceedings of Financial Cryptography and Data Security*.

[4] Muhammad Ahmad Bashir, Sajjad Arshad, Engin Kirda, William Robertson, and Christo Wilson. 2018. How Tracking Companies Circumvented Ad Blockers Using WebSockets. In *Proceedings of the ACM Internet Measurement Conference*.

[5] Muhammad Ahmad Bashir, Sajjad Arshad, Wil Robertson, Engin Kirda, and Christo Wilson. 2019. A Longitudinal Analysis of the ads.txt Standard. In *Proceedings of the ACM Internet Measurement Conference*.

[6] Muhammad Ahmad Bashir, Sajjad Arshad, William Robertson, and Christo Wilson. 2016. Tracing Information Flows Between Ad Exchanges Using Retargeted Ads. In *Proceedings of the USENIX Security Symposium*.

[7] Muhammad Ahmad Bashir and Christo Wilson. 2018. Diffusion of User Tracking Data in the Online Advertising Ecosystem. *Proceedings on Privacy Enhancing Technologies* 2018, 4 (2018).

[8] California Department of Justice 2020. Chapter 20. California Consumer Privacy Act Regulations. https://govt.westlaw.com/calregs/Browse/Home/California/CaliforniaCodeofRegulations?guid=IEB210D8CA2114665A08AF8443F0245AD&originationContext=documenttoc&transitionType=Default&contextData=(sc.Default).

[9] California State Legislature 2018. AB-375 California Consumer Privacy Act of 2018. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375.

[10] California State Legislature 2018. SB-1121 California Consumer Privacy Act of 2018. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB1121.

[11] Californians for Consumer Privacy Rights 2021. Annotated Text of the California Privacy Rights Act. https://www.caprivacy.org/annotated-cpra-text-with-ccpa-changes/.

[12] Jan Charatan and Eleanor Birrell. 2024. Two Steps Forward and One Step Back: The Right to Opt-out of Sale under CPRA. *Proceedings on Privacy Enhancing Technologies* 2024, 2 (2024), 91–105.

[13] Chrome Debugging Protocol 2023. Chrome DevTools Protocol Viewer. Chrome Dev. https://developer.chrome.com/devtools/docs/debugger-protocol.

[14] Lorrie Faith Cranor. 2012. Necessary but not Sufficient: Standardized Mechanisms for Privacy Notice and Choice. *Journal On Telecomm. & High Tech. L* 10, 2 (2012), 273–308.

[15] Adrian Dabrowski, Georg Merzdovnik, Johanna Ullrich, Gerald Sendera, and Edgar Weippl. 2019. Measuring Cookies and Web Privacy in a Post-GDPR World. In *Proceedings of the Passive and Active Measurement Conference*.

[16] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. 2019. We Value Your Privacy... Now Take Some Cookies: Measuring the GDPR's Impact on Web Privacy. In *Proceedings of the Network and Distributed System Security Symposium*.

[17] IAB Europe. 2023. FAQ: APD Decision on IAB Europe and TCF - Updated February 2023. IAB Europe. https://iabeurope.eu/wp-content/uploads/FAQ_-APD-DECISION-ON-IAB-EUROPE-AND-TCF-Updated-Febraury-2023.docx.pdf.

[18] Europe Protect Data Protection 2018. GDPR Consent Guidelines. European Data Protection Board. https://ec.europa.eu/newsroom/article29/items/623051/en.

[19] Federal Trade Commission 1998. Privacy Online: A Report to Congress. https://www.ftc.gov/sites/default/files/documents/reports/privacy-online-report-congress/priv-23a.pdf.

[20] Andrew Folks. 2023. US State Privacy Legislation Tracker. International Association of Privacy Professionals Resource Center. https://iapp.org/resources/article/us-state-privacy-legislation-tracker/.

[21] Imane Fouad, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic. 2020. Missed by Filter Lists: Detecting Unknown Third-Party Trackers with Invisible Pixels. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (2020), 499–518.

[22] GDPR 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation). OJ L 119 4.5.2016. , 88 pages.

[23] GDPR TCF Framework 2023. IAB Europe Transparency & Consent Framework 2.2. IAB Europe. https://github.com/InteractiveAdvertisingBureau/GDPR-Transparency-and-Consent-Framework/blob/master/TCFv2/IAB%20Tech%20Lab%20-%20CMP%20API%20v2.md.

[24] Global Privacy Control 2023. Global Privacy Control — Interacting With The GPC Signal — global-privacy-control. https://global-privacy-control.glitch.me/.

[25] Global Privacy Platform 2023. IAB Global Privacy Platform. IAB US. https://iabtechlab.com/gpp/.

[26] Johanna Gunawan, Amogh Pradeep, David Choffnes, Woodrow Hartzog, and Christo Wilson. 2021. A Comparative Study of Dark Patterns Across Mobile and Web Modalities. *Proceedings of the ACM: Human-Computer Interaction* 5, CSCW2 (2021).

[27] Hana Habib, Yixin Zou, Aditi Jannu, Neha Sridhar, Chelse Swoopes, Alessandro Acquisti, Lorrie Faith Cranor, Norman Sadeh, and Florian Schaub. 2019. An Empirical Analysis of Data Deletion and Opt-Out Choices on 150 Websites. In *Proceedings of the Workshop on Usable Security*.

[28] Maximilian Hils, Daniel W. Woods, and Rainer Böhme. 2020. Measuring the Emergence of Consent Management on the Web. In *Proceedings of the ACM Internet Measurement Conference*. 317–332.

[29] Maximilian Hils, Daniel W. Woods, and Rainer Böhme. 2021. Privacy Preference Signals: Past, Present and Future. *Proceedings on Privacy Enhancing Technologies* 2021, 4 (2021), 249–269.

[30] Henry Hosseini, Christine Utz, Martin Degeling, and Thomas Hupperich. 2024. A Bilingual Longitudinal Analysis of Privacy Policies Measuring the Impacts of the GDPR and the CCPA/CPRA. *Proceedings on Privacy Enhancing Technologies* 2024, 2 (2024), 434–463.

[31] IAB Tech Lab 2021. US Privacy String. IAB US. https://github.com/InteractiveAdvertisingBureau/USPrivacy/blob/master/CCPA/US%20Privacy%20String.md.

[32] IAB Tech Lab 2023. IAB CCPA Compliance Framework For Publishers & Technology Companies. IAB Technology Laboratory. https://iabtechlab.com/standards/ccpa/.

[33] Inmobi Acquired Quantcast 2023. Inmobi Acquired Quantcast. Inmobi. https://www.quantcast.com/press-release/inmobi-acquires-quantcast-choice-to-enhance-frictionless-consent-management-for-publishers/.

[34] Javascript Flags 2022. Property Flags and Descriptors. Javascript Info. https://javascript.info/property-descriptors.

[35] Tobias Lauinger, Abdelberi Chaabane, Sajjad Arshad, William Robertson, Christo Wilson, and Engin Kirda. 2017. Thou Shalt Not Depend on Me: Analysing the Use of Outdated JavaScript Libraries on the Web. In *Proceedings of the Network and Distributed System Security Symposium*.

[36] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proceedings of the Network and Distributed System Security Symposium*.

[37] Pedro Leon, Blase Ur, Richard Shay, Yang Wang, Rebecca Balebako, and Lorrie Cranor. 2012. Why Johnny Can't Opt out: A Usability Evaluation of Tools to Limit Online Behavioral Advertising. In *Proceedings of ACM Human Factors in Computing Systems*.

[38] Limited Service Provider 2019. IAB Limited Service Provider Agreement. IAB US. https://www.iabprivacy.com/lspa-2019-12.pdf.

[39] Zengrui Liu, Umar Iqbal, and Nitesh Saxena. 2024. Opted Out, Yet Tracked: Are Regulations Enough to Protect Your Privacy? *Proceedings on Privacy Enhancing Technologies* 2024, 1 (2024), 280–299.

[40] Celestin Matte, Nataliia Bielova, and Cristiana Santos. 2020. Do Cookie Banners Respect my Choice? Measuring Legal Compliance of Banners from IAB Europe's Transparency and Consent Framework. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[41] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. 2020. Dark Patterns after the GDPR: Scraping Consent Pop-Ups and Demonstrating Their Influence. In *Proceedings of ACM Human Factors in Computing Systems*.

[42] Sean O'Connor, Ryan Nurwono, Aden Siebel, and Eleanor Birrell. 2021. (Un)Clear and (In)Conspicuous: The Right to Opt-out of Sale under CCPA. In *Proceedings of the Workshop on Workshop on Privacy in the Electronic Society*.

[43] Lukasz Olejnik, Minh-Dung Tran, and Claude Castelluccia. 2014. Selling Off Privacy at Auction. In *Proceedings of the Network and Distributed System Security Symposium*.

[44] Page Evaluate Method 2017. Page Evaluate Method Chrome Dev Tools. Chrome. https://chromedevtools.github.io/devtools-protocol/tot/Page/#method-addScriptToEvaluateOnNewDocument.

[45] People of the State of California v. Sephora USA, Inc. 2022. Attorney General Bonta Announces Settlement with Sephora as Part of Ongoing Enforcement of California Consumer Privacy Act. State of California Department of Justice Press Release. https://oag.ca.gov/news/press-releases/attorney-general-bonta-announces-settlement-sephora-part-ongoing-enforcement.

[46] Privacy GPC 2022. Global Privacy Control. Global Privacy Control. https://privacycg.github.io/gpc-spec/.

[47] Iskander Sanchez-Rola, Matteo Dell'Amico, Platon Kotzias, Davide Balzarotti, Leyla Bilge, Pierre-Antoine Vervier, and Igor Santos. 2019. Can I Opt Out Yet?: GDPR and the Global Illusion of Cookie Control. In *Proceedings of the ACM ASIA Conference on Computer and Communications Security*.

[48] Aaron Sankin and Surya Mattu. 2020. The High Privacy Cost of a "Free" Website. https://themarkup.org/blacklight/2020/09/22/blacklight-tracking-advertisers-digital-privacy-sensitive-websites.
[49] Asuman Senol, Gunes Acar, Mathias Humbert, and Frederik Zuiderveen Borgesius. 2022. Leaky Forms: A Study of Email and Password Exfiltration Before Form Submission. In *Proceedings of the USENIX Security Symposium*. 1813–1830.
[50] Michael Smith, Antonio Torres-Agüero, Riley Grossman, Pritam Sen, Yi Chen, and Cristian Borcea. 2024. A Study of GDPR Compliance under the Transparency and Consent Framework. In *Proceedings of The Web Conference*. 1227–1236.
[51] Jannick Sørensen and Sokol Kosta. 2019. Before and After GDPR: The Changes in Third Party Presence at Public and Private European Websites. In *Proceedings of The Web Conference*.
[52] The European Parliament and the Council of the European Union 2002. Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications).
[53] Christof Ferreira Torres, Fiona Willi, and Shweta Shinde. 2023. Is Your Wallet Snitching On You? An Analysis on the Privacy Implications of Web3. *arXiv preprint arXiv:2306.08170* (2023).
[54] TRC 2020. DuckDuckGo Tracker Radar Collector. DuckDuckGo. https://github.com/duckduckgo/tracker-radar-collector.
[55] Martino Trevisan, Stefano Traverso, Eleonora Bassi, and Marco Mellia. 2019. 4 Years of EU Cookie Law: Results and Lessons Learned. *Proceedings on Privacy Enhancing Technologies* 2019, 2 (2019), 126–145.
[56] Tobias Urban, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2020. Beyond the Front Page: Measuring Third Party Dynamics in the Field. In *Proceedings of The Web Conference*. 1275–1286.
[57] Christine Utz, Sabrina Amft, Martin Degeling, Thorsten Holz, Sascha Fahl, and Florian Schaub. 2023. Privacy Rarely Considered: Exploring Considerations in the Adoption of Third-Party Services by Websites. *Proceedings on Privacy Enhancing Technologies* 2023, 1 (2023), 5–28.
[58] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. 2019. (Un)informed Consent: Studying GDPR Consent Notices in the Field. In *Proceedings of the ACM Conference on Computer and Communications Security*.
[59] Pelayo Vallina, Victor Le Pochat, Álvaro Feal, Marius Paraschiv, Julien Gamba, Tim Burke, Oliver Hohlfeld, Juan Tapiador, and Narseo Vallina-Rodriguez. 2020. Mis-shapes, Mistakes, Misfits: An Analysis of Domain Classification Services. In *Proceedings of the ACM Internet Measurement Conference*. 598–618.
[60] Maggie Van Nortwick and Christo Wilson. 2022. Setting the Bar Low: Are Websites Complying With the Minimum Requirements of the CCPA? *Proceedings on Privacy Enhancing Technologies* 2022, 1 (2022), 608–628.
[61] Sebastian Zimmeck, Eliza Kuller, Chunyue Ma, Bella Tassone, and Joe Champeau. 2024. Generalizable Active Privacy Choice: Designing a Graphical User Interface for Global Privacy Control. *Proceedings on Privacy Enhancing Technologies* 2024, 1 (2024), 258–279.
[62] Sebastian Zimmeck, Oliver Wang, Kuba Alicki, Jocelyn Wang, and Sophie Eng. 2023. Usability and Enforceability of Global Privacy Control. *Proceedings on Privacy Enhancing Technologies* 2004, 2 (2023), 1–17.