Browsing without Third-Party Cookies: What Do You See?

Maxwell Lin
Duke University

Shihan Lin
Duke University

Helen Wu
Vanderbilt University

Karen Wang
Duke Kunshan University

Xiaowei Yang
Duke University

ABSTRACT

Web cookies are often used for privacy-invasive behavior tracking. Previous studies have proposed methods to automatically disable privacy-invasive cookies, either by interacting with cookie notices or filtering out cookies at the browser-level. In this study, we advocate for complete third-party cookieless browsing. We develop a framework to analyze the impact of removing third-party cookies on website rendering and implement this framework in a Selenium-based web crawler to measure the top 10,000 Tranco domains. We find that disabling third-party cookies has no significant effect on website appearance, text, images, or links. This validates the industry-wide shift towards cookieless browsing as a way to protect user privacy without compromising on the user experience.

1 INTRODUCTION

Websites use web cookies to record stateful information in a browsing session. While some cookies are necessary for websites to work properly (e.g., authentication cookies), the majority of cookies are used for user tracking and advertising [21]. In a web measurement study, Englehardt et al. found that an adversary can reconstruct 62-73% of a typical user's browsing history using third-party cookies [13].

Because of general privacy concerns, many governments have enacted regulations that require websites to implement cookie notices. These cookie notices allow users to consent to or opt out of the use of unnecessary cookies. For example, the California Consumer Privacy Act (CCPA) [8] is a regulation that requires websites to provide a clear opt-out mechanism for their users. In the European Union, the General Data Protection Regulation (GDPR) [15] and ePrivacy Directive [14] requires websites to obtain specific, informed, and unambiguous user consent before accessing or storing any user data unessential to website function. Previous studies have measured the effect of privacy regulations on the cookie landscape. Degeling et al. discovered that the use of cookie notices increased by 16% in EU member states after GDPR went into effect [12]. Furthermore, cookie regulations have given rise to Consent Management Platforms (CMPs) which provide "consent as a service" solutions to websites. Due to their ease of use, many websites use CMPs to implement their cookie notices as well as to manage and disseminate user consent to third parties.

Although cookie notices are intended to give users more control over their privacy, 88% of cookie notices incorporate design techniques that hinder the ability of users to select privacy-protective options [19]. In a user study by Habib et al., it was found that 73% of participants opted for the most permissive cookie settings and only 45% of participants chose the settings that genuinely reflected their desired level of consent [19]. This result indicates that cookie notices are an unreliable way for users to choose their desirable privacy settings.

As a potential solution, *notice-centric* methods have been proposed to automate user-interaction with cookie notices. For example, Khandelwal et al. developed *CookieEnforcer*, a browser extension which can automatically disable all unnecessary cookies with 94% accuracy [23]. However, notice-centric methods assume that websites will honor the choices that users make, which may not be the case in practice.

In a preliminary study of 255 websites that implemented cookie notices, we found that 39% of them do not respect the user's choice and continue to use tracking cookies even after the user opts out in the provided cookie notice (see Figure 1). Even websites that employ CMPs violate cookie regulations. We crawled 118 websites that use *OneTrust*, the most popular CMP, and we found that 58% use tracking cookies even when they are disabled via the OneTrust API. Lastly, by design, notice-centric methods cannot work if the website does not provide a cookie notice in the first place. In a large scale study of over 17,000 websites, Kampanos et al. found that less than 50% of websites show a cookie notice to the user [22]. Therefore, notice-centric cookie enforcement will be unsuccessful on a majority of websites.

Due to limitations in the notice-centric approach, we propose to directly intercept cookies at the browser level. We seek to create a policy framework that enables a user to decide whether a cookie should be enabled without depending on whether a cookie notice exists on the website at all. For example, a privacy-conscious user may want to enable only strictly necessary cookies. However, this approach requires a cookie classifier, and as we explain in Section 2.0.2, categorizing cookies by function is difficult.

Partly due to the abovementioned challenges of disabling *all* privacy-invasive cookies, a recent trend across web browsers has been to disable *all* third-party cookies. This approach is easy to implement and requires little maintenance (compared to categorizing cookies by functions and disabling privacy-invasive ones). Google Chrome, the most popular web browser, is following this approach and will deprecate all third-party cookies in early 2025 [18]. Other popular browsers such as Firefox [3] and Safari [1] already block many third-party cookies by default. We refer to this approach as *cookieless browsing*.

While cookieless browsing certainly limits websites from tracking users, it remains unclear whether disabling all third-party cookies impacts the user experience of websites. This ambiguity motivates our present study, where we design and implement a framework to rigorously assess the influence of cookieless browsing on key website operations, such as layout rendering, text/image content, and interactivity. We seek to answer the question: *How does third-party cookieless browsing affect the way websites are displayed to users?*

1.0.1 Contributions. We design a framework to measure how removing third-party cookies impacts website rendering. To our knowledge, this is the first large-scale study analyzing this behavior. We implement this framework using a Selenium-based web crawler and deploy it across the top 10,000 Tranco domains. By analyzing extracted features (screenshots, text, images, and links) across different crawl groups, we find that the absence of third-party cookies does not significantly affect the rendering of websites. More than 90% of the domains showed minimal changes in website rendering, suggesting that cookieless browsing will not degrade user experience. Lastly, we release our implementation and results for reproducibility and further analysis.

1.0.2 Organization. The remainder of the paper is structured as follows. Section 2 evaluates related studies. Section 3 describes our framework and implementation considerations. Section 4 describes our approach to comparing extracted features. Section 5 presents the results of our large-scale crawl. Section 6 evaluates our study and outlines future work. Section 7 draws conclusions.

2 BACKGROUND & RELATED WORK

In this section, we provide definitions and evaluate related studies.

2.0.1 ICC UK Cookie Categories. There are four different types of cookies defined by the International Chamber of Commerce UK [10]:

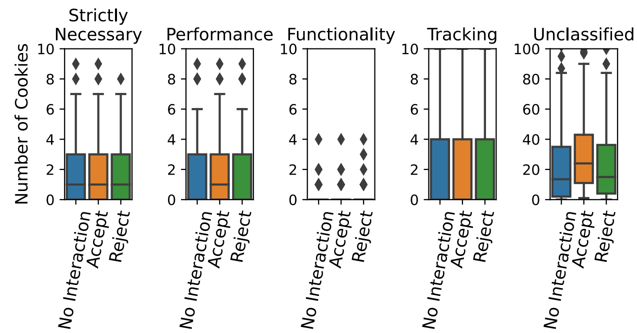


Figure 1: Cookie-Script Categorization for Different Interaction Modes. Cookies were collected from 255 websites that implemented cookie notices. *BannerClick* [27] was used to click the accept and reject buttons. Note that *Cookie-Script* was unable to classify the vast majority of cookies collected; there are roughly ten times more unclassified cookies than any of the four categorized cookie types. (Most websites had zero functionality cookies; outliers are shown as diamonds.)

(1) **Strictly Necessary Cookies:** Enable users to move around the website and use requested features, such as accessing secure areas of the website or adding items to a shopping cart.
(2) **Performance Cookies:** Collect anonymized information about how visitors use a website (e.g., popular pages, error logs).
(3) **Functionality Cookies:** Remember choices that users make (such as username, language, or region) to provide personalized features.
(4) **Tracking Cookies:** Collect information about users' browsing habits to deliver relevant advertisements.

2.0.2 Automated Cookie Classification. There are many cookie databases that attempt to map cookies to their ICC UK category. For example, when given a website, *Cookie-Script* [11] will categorize all present cookies as either one of the four ICC UK categories or the *unclassified* category if no database entry is found. In a preliminary study, we crawled 255 websites and categorized collected cookies using *Cookie-Script*. We found that the vast majority of cookies were unable to be classified (see Figure 1). Due to the ever-changing nature of the web, cookie databases will never be fully comprehensive.

A potential solution to this problem is to train a machine learning classifier. Hu et al. [21] introduced *CookieMonster*, a machine learning model capable of categorizing cookies with 94% accuracy. Bollinger et al. [6] developed *CookieBlock*, a machine-learning classifier and browser extension which automatically blocks privacy-invasive cookies with 90% accuracy. Unfortunately, machine-learning models are never completely accurate, and such methods may still be unsatisfactory to a particularly privacy-conscious user.

3 WEB CRAWLER ARCHITECTURE

Our crawler uses Selenium [28] to drive a headless Firefox instance. We deploy our crawler on the top 10,000 domains of the `KJ2GW` Tranco [24] list generated on 18 February 2024. [1] The total running time across all jobs was 241 days; the median time to crawl one domain was 33 minutes. If a domain took more than 60 minutes, we terminated the process and moved on to the next domain.

3.1 Domain to URL Resolution

We first resolve the apex domain obtained from Tranco to an accessible URL. An apex domain is a two-level domain [20], e.g., `example.com`. To resolve the apex domain, we alternate between the `http` and `https` protocols in addition to optionally prepending the `www` subdomain. For example, to resolve the example.com domain, we attempt navigation to https://example.com, https://www.example.com, http://example.com, and http://www.example.com. Domain to URL resolution is successful if we can retrieve a web page with clickable elements (see Section 3.3).

3.2 Crawl Groups

To measure the effects of third-party cookies on website behavior, we crawl each domain three separate times corresponding to the following groups:

(1) **Baseline:** We generate and save a length 5 clickstream with all cookies enabled. See Section 3.3.
(2) **Control:** We traverse the generated clickstream under the same conditions as the *baseline* group. See Section 3.4.
(3) **Experimental:** We traverse the generated clickstream with all third-party cookies disabled. See Section 3.4.

A unique Firefox profile is created for each group to isolate stateful information such as cookies between browsing sessions.

The motivation behind these three groups is as follows: if we observe a significant difference between the *baseline* and *experimental* groups, it is possible that the website requires the use of cookies. However, we cannot conclude that the experimental condition caused this change in behavior since the content of some websites can change naturally upon page reload (e.g., social media websites). To take this into account, we crawl each website twice without applying the experimental condition (i.e., the *baseline* and *control* groups). If these two groups are similar but the experimental group

[1] Available at https://tranco-list.eu/list/KJ2GW.

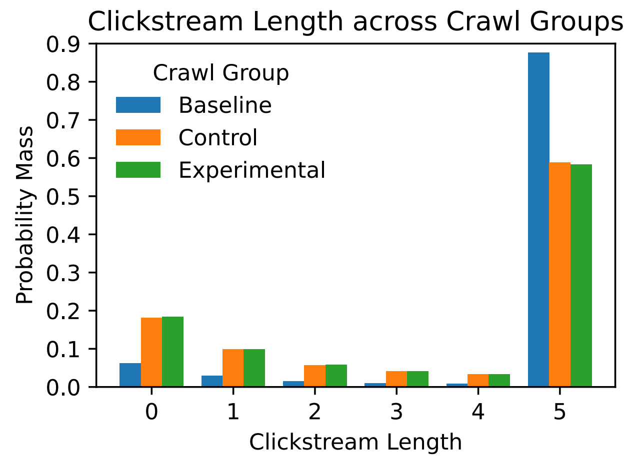


Figure 2: Clickstream Length across Crawl Groups. Clickstream generation occurs in the baseline group while clickstream traversal occurs in the control and experimental groups. The average generated clickstream length is 4.5, while the average traversed clickstream length is 3.4. The success rate of generating a 5 length clickstream is about 0.9, while the success rate of traversing a 5 length clickstream is only about 0.6.

is significantly different, then we can conclude that the experimental condition (disabling third-party cookies) likely caused the observed change in website behavior.

3.3 Clickstream Generation

A length k clickstream is a list of k actions sampled from a set of clickable elements. We define a clickable element as an HTML element that satisfies at least one of the following criteria:

- **Button:** Elements with a `<button>` tag. [2]
- **Link:** Elements with an `<a>` tag. [3]
- **Onclick:** Elements with an `onclick` event attribute. [4]
- **Pointer:** Elements with a "pointer" `cursor` style. [5]

See Figure 3 for the percentage of clickable element types encountered during clickstream generation.

After navigating to the landing page, we construct a set of clickable elements. We sample uniformly at random from this set until a successful click is made. This *action* is then appended to the current clickstream and the set of clickable elements is reconstructed. This process continues until the

[2] See https://developer.mozilla.org/en-US/docs/Web/HTML/Element/button.
[3] See https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a.
[4] See https://developer.mozilla.org/en-US/docs/Web/API/Element/click_event.
[5] See https://developer.mozilla.org/en-US/docs/Web/CSS/cursor.

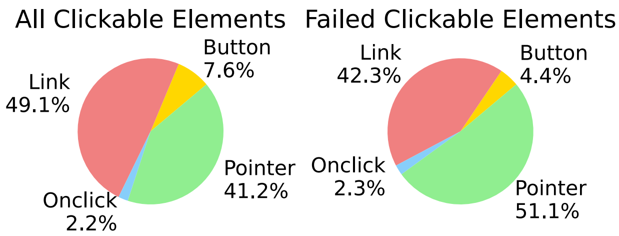


Figure 3: Clickable Elements. The first plot displays the percentage of clickable elements encountered during clickstream generation. The second plot displays the percentage of clickable elements that failed during clickstream traversal. A clickable element fails when its CSS selector is unable to be resolved.

desired clickstream length is generated. If the set of clickable elements is ever empty or we navigate to a different domain, the clickstream prematurely ends.

Ideally, the clickstream length should be long enough such that the inner pages of the website are thoroughly explored. However, longer clickstreams become more difficult to fully traverse as more actions must be successfully completed sequentially. By analyzing the browsing data of 2.5M users across 760 websites, Lehmann et al. found that users only view 2.36 pages of a website on average [25]. Xing et al. examined how users navigate search engine results and found that the majority of users conduct less than 2 clicks per session [31]. Thus, we selected a clickstream length of 5 which we found to achieve a good balance between inner page exploration and traversal success.

In our implementation, clickstream actions are saved using a unique CSS selector generated by the finder [26] JavaScript library. According to an independent benchmark [17], finder was able to consistently generate the shortest selectors compared to 12 other libraries. Shorter selectors are more reliable across page visits, resulting in more successful clickstream traversals.

3.4 Clickstream Traversal

After navigating to the landing page, we execute each action in the clickstream sequentially. Sometimes, the CSS selector originally generated cannot be resolved and we are unable to fully traverse the clickstream. Figure 2 presents the normalized histogram of clickstream lengths across crawl groups. Note that making the clickstream longer reduces the number of successful full-length clickstream traversals. See Figure 3 for the percentage of clickable element types that failed during traversal.

3.5 Extracted Features

After navigating to the landing page and after each action in a clickstream, we extract the following features:

1. **Screenshot:** We take a screenshot of the viewport after scrolling to the top of the page. This ensures that screenshots are consistently aligned.
2. **Image Shingles:** We also convert the screenshot to an image shingle frequency vector (see Section 3.5.1).
3. **Inner Text:** The `innerText` of the `<body>` element [6] extracted as a word frequency vector.
4. **Images:** The `src` of all `<img>` elements [7] extracted as a frequency vector.
5. **Links:** The `href` of all links [8] present extracted as a frequency vector.

For each domain, we attempt to collect up to 50 comparable sets of features.

3.5.1 Image Shingle Extraction. Image shingling, proposed by Anderson et al., [4] is based on the notion of shingling from the text similarity literature. First, we divide each image into fixed sized chunks in memory. Like Anderson et al., we found that an image chunk size of 40x40 pixels was an effective trade-off between granularity and shingling performance. Each chunk is then hashed using MD5 to form an image shingle. Lastly, we extract the frequency vector of image shingles.

4 ANALYSIS

4.1 Screenshot Comparison

To compare baseline, control, and experimental screenshots (Feature 1), we use Algorithm 1 to obtain a difference $\Delta \in [0, 1]$ that accounts for dynamic content.

Concretely, we split each crawl group screenshot into 40x40 pixel chunks. Similar to image shingling, we found that this chunk size offers good balance between both precision and performance. We filter out all chunks that differ between the baseline and control group to ensure that any naturally occurring differences are excluded. For all remaining chunks, we compute the proportion that differ between the baseline and experimental group. This allows us to only measure differences that occur due to the experimental condition. Note that if baseline and control are identical, we simply return the percent difference between baseline and experimental.

If there are no chunks remaining after the filter (i.e., the baseline and control screenshots are completely different), then we skip the comparison.

[6]See https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/innerText.
[7]See https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img.
[8]See https://developer.mozilla.org/en-US/docs/Web/API/Document/links.

Algorithm 1 BCE Screenshot Difference

▷ chunk returns a list of chunks given a baseline screenshot, a control screenshot, and an experimental screenshot.
initialize chunks ← chunk(screenshots to compare)

initialize matches ← 0
initialize total ← 0
▷ Loop over crawl group chunks in parallel.
for each (baseline, control, experimental) in chunks **do**
    if baseline == control **then**
        total ← total + 1
        if baseline == experimental **then**
            matches ← matches + 1

▷ Return a difference $\Delta \in [0, 1]$.
return $1 - \left(\frac{\text{matches}}{\text{total}}\right)$

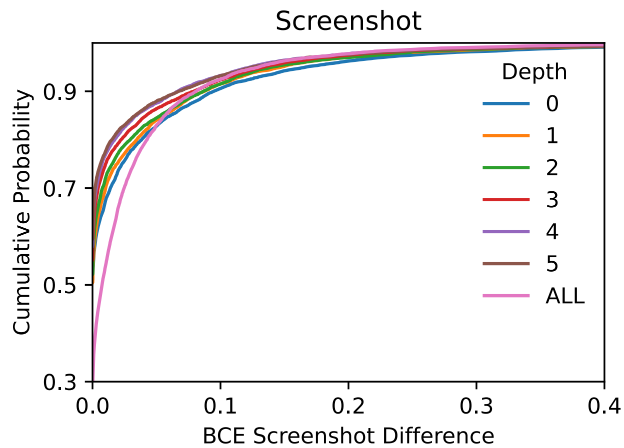


Figure 4: CDF of BCE Screenshot Differences. The BCE screenshot difference statistic is outlined in Algorithm 1.

4.2 Frequency Vector Comparison

Extracted frequency vectors (Features 2-5) are compared using the Jaccard distance formula. If we let A and B be the multisets of two frequency vectors, we define the Jaccard distance as

$$J(A, B) := 1 - \frac{|A \cap B|}{|A \cup B|} \in [0, 1]. \quad (1)$$

If $|A \cup B| = 0$, i.e., A and B are both empty, then we define $J(A, B) := 0$. In other words, we consider two empty sets to be identical.

We compare the extracted features using a difference in difference approach. The difference in difference (DiD) computes the following

$$\text{DiD} := J(B, E) - J(B, C) \in [-1, 1] \quad (2)$$

where B, C, E denote a frequency vector from the baseline, control, and experimental groups respectively.

If the DiD is small, then the experimental condition did not have a substantial effect since $J(B, E) \approx J(B, C)$.

5 RESULTS

Out of the 10,000 input Tranco domains, 1,897 domains were unable to be resolved to a URL. Through manual inspection, we found that many of these domains (e.g., `akamai.net`) belong to content delivery networks. After filtering out other problematic domains (e.g. domains which hang our crawler), we arrive at a final list of 7,490 successfully crawled domains.

We first compare screenshots using Algorithm 1. The distribution of differences is plotted as a CDF in Figure 4. We find that more than 90% of domains exhibit less than a 10% screenshot difference when cookies are disabled. These results suggest that disabling all third-party cookies generally does not not affect the screenshots of websites.

We also compare the extracted frequency vectors. CDFs of frequency vector DiDs are plotted in Figure 5 and the respective histograms are plotted in Figure 6. The distribution is concentrated and symmetric at 0 with a low standard deviation. These results suggest that disabling all third-party cookies generally does not affect the content or layout of a website.

6 DISCUSSION

In our study, we modeled user web browsing behavior as a sequence of clickstreams. This approach allowed for the measurement of the inner pages of a website, a consideration that previous studies [5] have emphasized. This approach also facilitated the creation of multiple crawl groups that differed only in their experimental condition (i.e., whether cookies were enabled or disabled). However, there are two important limitations of this clickstream model:

(1) **Dynamic content:** Because many webpages are dynamic, generated clickstreams may not be stable across time. Therefore, our dataset is biased towards static inner pages which may cause us to underestimate the impact cookies have on website behavior.
(2) **Random traversal:** To generate a clickstream, we repeatedly sample uniformly at random from the current set of clickable elements. However, random traversal likely does not capture the intricacies of real-user behavior. Consider a typical e-commerce website with 100 clickable elements per page. For a length 5 clickstream, the

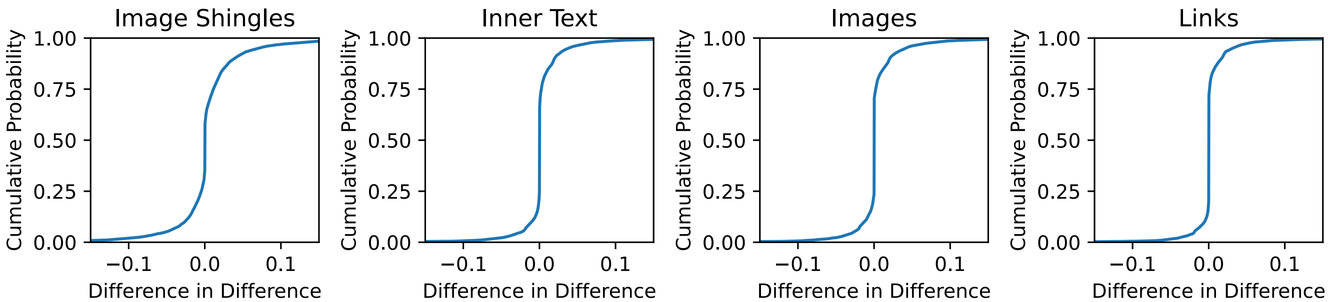


Figure 5: CDF of Frequency Vector DiDs. The difference in difference statistic is defined in Equation 2. The distribution is concentrated and symmetric at 0.

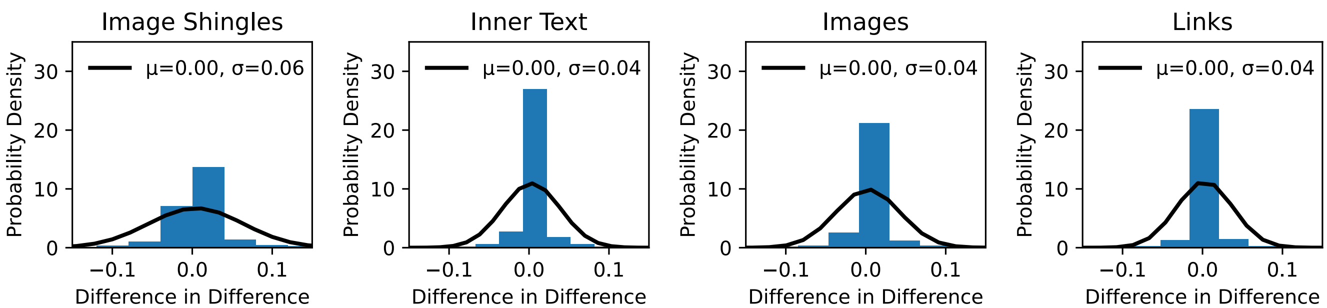


Figure 6: Histogram of Frequency Vector DiDs. The difference in difference statistic is defined in Equation 2. A fitted Gaussian is overlayed over each histogram. For all features, the fitted mean is 0.

state space size can be upper bounded by 100^5. Clearly, it is computationally infeasible to exhaustively explore this search space. To model *purposeful* web browsing, we must use a more targeted clickstream generation method. Unfortunately, generating targeted clickstreams is a difficult problem. White et al. has shown that there is high variability in the behavior of search engine users in their issued query, clicked result, and post-query browsing [29, 30]. This implies that there is no "average" user that can be modeled.

Our results suggest that third-party cookies do not have a significant effect on the behavior of most websites. Instead of relying on notice-centric cookie enforcement or cookie blocklists, a user can simply disable *all* third-party cookies at the browser-level without consequences for most websites.

This provides validation to the industry-wide shift towards cookieless browsing. Users will likely observe little to no change as browsers begin to disable all third-party cookies by default. On the other hand, advertisers will either need to adopt cookie substitutions (such as the APIs proposed in Google's Privacy Sandbox) or move back to non-targeted advertising [2].

As the majority of third-party cookies are *tracking* cookies [7], disabling all third-party cookies will certainly improve user privacy. However, domains can still track users across websites using only *first-party* cookies, through techniques such as cookie respawning [16] or first-party cookie leakage [9]. Thus, a natural extension of the present study is to examine how disabling *all* web cookies affects website behavior.

7 CONCLUSION

In this study, we develop a framework to determine whether a given website requires the use of cookies. Specifically, we devise different groups which crawl the same clickstream with varying experimental conditions. We implement this framework in a Selenium-based web crawler and deploy it across the top 10,000 Tranco domains. By comparing extracted screenshots, text, images, and links, we conclude that third-party cookies do not have an observable impact on the vast majority of domains.

REFERENCES

[1] 2020. Tracking Prevention in WebKit. https://webkit.org/tracking-prevention/

[2] 2024. The Future of Targeted Advertising in a Cookie-less World. https://insight.kellogg.northwestern.edu/article/online-consumer-privacy-advertising

[3] 2024. Third-party cookies and Firefox tracking protection | Firefox Help. https://support.mozilla.org/en-US/kb/third-party-cookies-firefox-tracking-protection#

[4] David S. Anderson, Chris Fleizach, Stefan Savage, and Geoffrey M. Voelker. 2007. Spamscatter: Characterizing Internet Scam Hosting Infrastructure. https://www.usenix.org/conference/16th-usenix-security-symposium/spamscatter-characterizing-internet-scam-hosting

[5] 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*. ACM, Virtual Event USA, 680–695. https://doi.org/10.1145/3419394.3423626

[6] Dino Bollinger, Karel Kubicek, Carlos Cotrini, and David Basin. 2022. Automating Cookie Consent and {GDPR} Violation Detection. 2893–2910. https://www.usenix.org/conference/usenixsecurity22/presentation/bollinger

[7] Aaron Cahn, Scott Alfeld, Paul Barford, and S. Muthukrishnan. 2016. An Empirical Study of Web Cookies. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, Montréal Québec Canada, 891–901. https://doi.org/10.1145/2872427.2882991

[8] CCPA 2018. California Consumer Privacy Act (CCPA). https://oag.ca.gov/privacy/ccpa.

[9] Quan Chen, Panagiotis Ilia, Michalis Polychronakis, and Alexandros Kapravelos. 2021. Cookie Swap Party: Abusing First-Party Cookies for Web Tracking. In *Proceedings of the Web Conference 2021 (WWW '21)*. Association for Computing Machinery, New York, NY, USA, 2117–2129. https://doi.org/10.1145/3442381.3449837

[10] Cookie Guide 2012. ICC UK Cookie Guide. https://www.cookielaw.org/wp-content/uploads/2019/12/icc_uk_cookiesguide_revnov.pdf.

[11] CookieScript 2024. Cookie-Script. https://cookie-script.com/.

[12] 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 2019 Network and Distributed System Security Symposium*. Internet Society, San Diego, CA. https://doi.org/10.14722/ndss.2019.23378

[13] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan Mayer, Arvind Narayanan, and Edward W. Felten. 2015. Cookies That Give You Away: The Surveillance Implications of Web Tracking. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, Florence Italy, 289–299. https://doi.org/10.1145/2736277.2741679

[14] ePrivacy Directive 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). https://data.europa.eu/eli/dir/2002/58/oj/eng.

[15] European Parliament and Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council. https://data.europa.eu/eli/reg/2016/679/oj

[16] Imane Fouad, Cristiana Santos, Arnaud Legout, and Nataliia Bielova. 2021. Did I delete my cookies? Cookies respawning with browser fingerprinting. http://arxiv.org/abs/2105.04381 arXiv:2105.04381 [cs].

[17] Riki Fridrich. 2024. fczbkk/css-selector-generator-benchmark: benchmark for various JavaScript libraries for generating CSS selectors. https://github.com/fczbkk/css-selector-generator-benchmark

[18] Google. 2024. Prepare for Phasing out Third-Party Cookies | Privacy Sandbox. https://developers.google.com/privacy-sandbox/3pcd.

[19] Hana Habib, Megan Li, Ellie Young, and Lorrie Cranor. 2022. "Okay, whatever": An Evaluation of Cookie Consent Interfaces. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–27. https://doi.org/10.1145/3491102.3501985

[20] Paul E. Hoffman, Anew Sullivan, and Kazunori Fujiwara. 2019. *DNS Terminology.* Request for Comments RFC 8499. Internet Engineering Task Force. https://doi.org/10.17487/RFC8499 Num Pages: 50.

[21] Xuehui Hu, Nishanth Sastry, and Mainack Mondal. 2021. CCCC: Corralling Cookies into Categories with CookieMonster. In *Proceedings of the 13th ACM Web Science Conference 2021 (WebSci '21)*. Association for Computing Machinery, New York, NY, USA, 234–242. https://doi.org/10.1145/3447535.3462509

[22] Georgios Kampanos and Siamak F. Shahandashti. 2021. Accept All: The Landscape of Cookie Banners in Greece and the UK. https://doi.org/10.48550/arXiv.2104.05750 arXiv:2104.05750 [cs].

[23] Rishabh Khandelwal, Asmit Nayak, Hamza Harkous, and Kassem Fawaz. 2023. Automated Cookie Notice Analysis and Enforcement. 1109–1126. https://www.usenix.org/conference/usenixsecurity23/presentation/khandelwal

[24] 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 26th Annual Network and Distributed System Security Symposium (NDSS 2019)*. https://doi.org/10.14722/ndss.2019.23386

[25] Janette Lehmann, Mounia Lalmas, Georges Dupret, and Ricardo Baeza-Yates. 2013. Online multitasking and user engagement. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management (CIKM '13)*. Association for Computing Machinery, New York, NY, USA, 519–528. https://doi.org/10.1145/2505515.2505543

[26] Anton Medvedev. 2024. antonmedv/finder. https://github.com/antonmedv/finder original-date: 2018-02-20T16:02:13Z.

[27] Ali Rasaii, Shivani Singh, Devashish Gosain, and Oliver Gasser. 2023. Exploring the Cookieverse: A Multi-Perspective Analysis of Web Cookies. In *Passive and Active Measurement*, Anna Brunstrom, Marcel Flores, and Marco Fiore (Eds.). Vol. 13882. Springer Nature Switzerland, Cham, 623–651. https://doi.org/10.1007/978-3-031-28486-1_26 Series Title: Lecture Notes in Computer Science.

[28] Selenium 2024. Selenium. https://www.selenium.dev/.

[29] Ryen W. White and Steven M. Drucker. 2007. Investigating behavioral variability in web search. In *Proceedings of the 16th international conference on World Wide Web (WWW '07)*. Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/1242572.1242576

[30] Ryen W. White and Dan Morris. 2007. Investigating the querying and browsing behavior of advanced search engine users. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '07)*. Association for Computing Machinery, New York, NY, USA, 255–262. https://doi.org/10.1145/1277741.1277787

[31] Qianli Xing, Yiqun Liu, Jian-Yun Nie, Min Zhang, Shaoping Ma, and Kuo Zhang. 2013. Incorporating user preferences into click models. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*. ACM Press, San Francisco, California, USA, 1301–1310. https://doi.org/10.1145/2505515.2505704

A APPENDIX

A.1 Ethics

This work does not raise any ethical issues.

A.2 Screenshot Examples

The following figures provide some examples of dynamic content and screenshot comparisons.

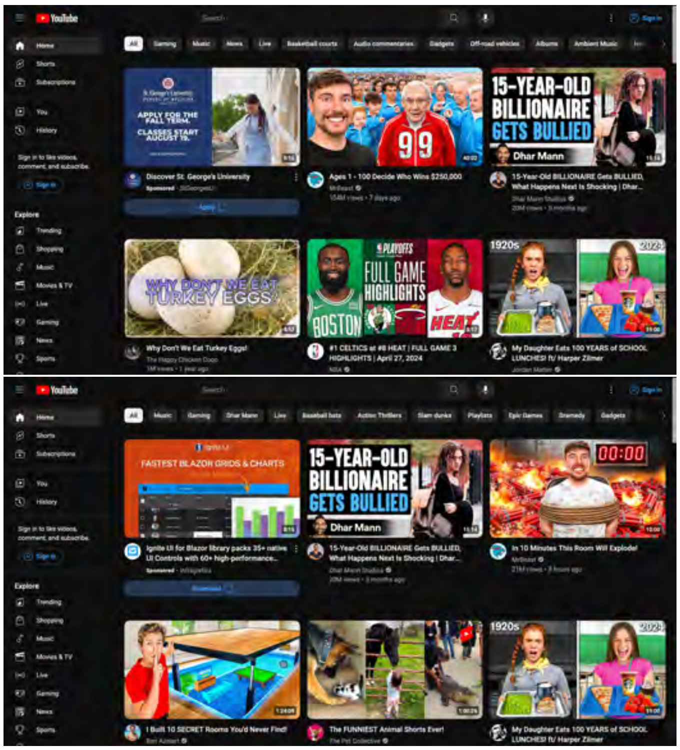
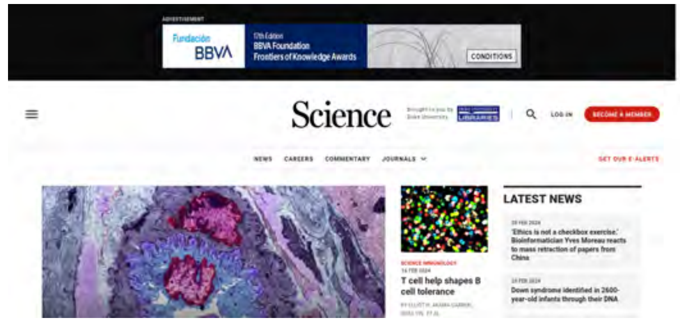
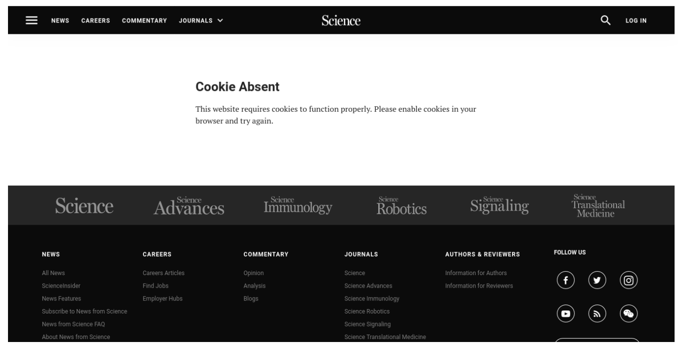


Figure 7: The website `youtube.com` displays dynamic content. These two screenshots show how `youtube.com` changes between page reloads.

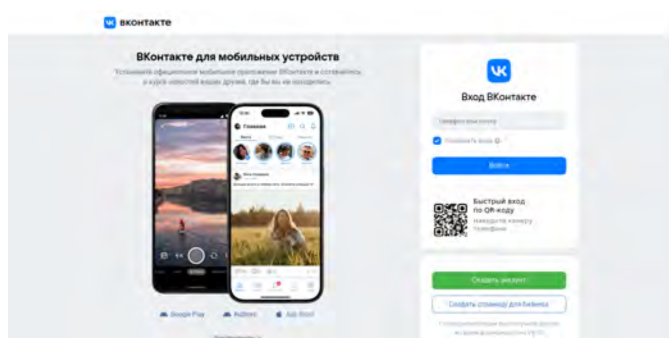


(a) Baseline



(b) Experimental

Figure 8: Without cookies, `sciencemag.org` displays an error message. There is a 0.85 jaccard distance between the image shingles of these two screenshots.



(a) Baseline

(b) Experimental

Figure 9: Without cookies, `userapi.com` is unable to change languages. There is a 0.35 jaccard distance between the image shingles of these two screenshots.