CSyncProxy: Differentially Private Third-party Cookie Synchronization

Sarah Bellemare
McGill University
Montreal, Canada

Daniel Migault
Ericsson Research
Montreal, Canada

Benjamin C. M. Fung
McGill University
Montreal, Canada

Stere Preda
Ericsson Research
Montreal, Canada

Amine Boukhtouta
Ericsson Research
Montreal, Canada

***Abstract*—Cookie synchronization enables multiple advertising networks to share user browsing data, thus refining ad targeting without explicit user consent. Although existing mitigation strategies, such as cookie blockers, reduce privacy risks, they can also decrease website revenue or restrict certain online services. To address this limitation, we introduce a differentially private cookie synchronization proxy, *CSyncProxy*, that leverages the exponential mechanism to grant users granular control over the amount of data shared during cookie synchronization. By obfuscating user identities across websites, the proxy maintains sufficient personalization for advertisements while safeguarding user privacy. Experimental web crawls of the top 100 websites per visit and analysis of the likelihood of successful anonymization indicate that our approach reduces the number of instances of cookie synchronization by over 40% compared to standard browsing and 11% compared to Chrome's cookie-blocking feature, without blocking or barring the user from any website.**

1. Introduction

As users browse the Internet, their history influences targeted ads, making ad space rental a major revenue source for websites [22]. Ad networks earn revenue from clicks on ads, which drives them to precisely tailor these ads. They commonly track user behavior through third-party cookies–set by external domains–often without user awareness. Third-party cookies are set by external domains (e.g., ad networks) through embedded content such as ads, scripts, or tracking pixels [1]. When a website includes a script from an advertising network, the browser requests content from that third-party domain, which then sets a cookie. On subsequent visits to other websites using the same ad network, the cookie is sent back, allowing the network to track browsing behavior across multiple sites.

Given the privacy concerns associated with extensive data collection, various regulatory measures have been introduced. The *General Data Protection Regulation* (*GDPR*) [24] imposes strict requirements on how organizations must implement cookies and obtain user consent. Consequently, websites are now required to secure explicit consent from users before setting optional cookies and are restricted to collecting only the minimum necessary data. In addition, the *Same-Origin Policy* (*SOP*) [25] prevents third-party domains from accessing or modifying resources belonging to other domains on the same webpage, effectively denying two third-party advertisements from directly sharing data with each other.

To circumvent these restrictions, many advertising networks participate in *cookie synchronization*, a process that allows them to share cookie values and thus accumulate a more comprehensive profile of the browsing history of a user. This approach not only maintains current datasets but also enhances their overall knowledge base for 97% of Internet users [16], leading to increased profitability in online behavioral advertising. Figure 1 illustrates cookie synchronization. When a user visits a website, their browser loads JavaScript from third-party trackers, which then set cookies. Ad Network 1 assigns a cookie ID (*ad1id*), stores it, and then redirects the browser to Ad Network 2, transmitting *ad1id*. Ad Network 2 stores this ID, sets its own cookie (*ad2id*), and links both IDs to the same user, allowing data aggregation between sites. As a result, cookie synchronization raises privacy concerns, as it typically occurs without explicit user consent. Therefore, mitigating the negative effects of cookie synchronization is crucial to ensure meaningful online anonymity.

In this paper, we propose a proxy-based solution, called *CSyncProxy*, that incorporates differential privacy to anonymize third-party cookies, reducing the impact of cookie synchronization while allowing users to adjust their level of privacy via a controllable privacy metric. The proxy dynamically tunes the degree of anonymization to balance privacy and utility. Finally, we present a study that evaluates the effectiveness of the proxy.

The method presented in this paper uses a secure browser architecture alongside the widespread deployment of HTTPS to mitigate cookie synchronization. Specifically, we propose a cookie anonymization proxy that employs local differential privacy techniques.

As this proxy intercepts user HTTPS traffic, it may initially appear similar to a *Man-in-the-Middle* (*MITM*) *attack*, potentially raising concerns about data monitoring and tracking. However, our design explicitly avoids prolonged data retention by not storing user information

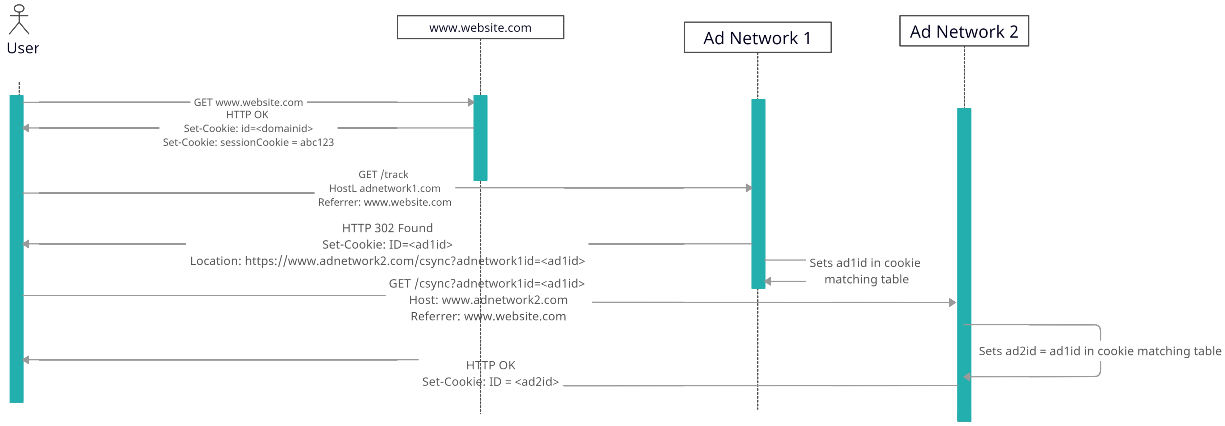
User
www.website.com
Ad Network 1
Ad Network 2
GET www.website.com
HTTP OK
Set-Cookie: id=<domainid>
Set-Cookie: sessionCookie = abc123
GET /track
HostL adnetwork1.com
Referrer: www.website.com
HTTP 302 Found
Set-Cookie: ID=<ad1id>
Location: https://www.adnetwork2.com/csync?adnetwork1id=<ad1id>
Sets ad1id in cookie matching table
GET /csync?adnetwork1id=<ad1id>
Host: www.adnetwork2.com
Referrer: www.website.com
Sets ad2id = ad1id in cookie matching table
HTTP OK
Set-Cookie: ID = <ad2id>

Figure 1: Client-ad platform HTTPS flow showing cookie synchronization

in the long term. In addition, we envision deploying the Cookie Proxy within a *Trusted Execution Environment* (*TEE*), which is attested to the end user. Through this combination of TEE and attestation, users can verify that the open-source audited code that executes within the TEE is precisely the one intended, thus preventing unverified or malicious code from running. The primary contributions of our work are as follows.

- *Local Differential Private Cookie Synchronization:* We introduce a local differential privacy method to mitigate the effects of cookie synchronization, leveraging the exponential mechanism to allow users to select their desired level of privacy. To our knowledge, this is the first application of local differential privacy to address cookie synchronization.
- *Novel Anonymization Proxy:* We present an anonymization proxy that acts as an intermediary between end users and cookie collection systems. By obfuscating the data that is shared cookie synchronization and employing the exponential mechanism, the proxy prevents third parties from exchanging user information.
- *Privacy-Usability Trade-off Control:* We demonstrate how users can tune privacy by adjusting the privacy budget in the exponential mechanism. We show experimentally how these user choices affect the probability of a successful cookie anonymization, enabling fine-grained control over user privacy.

2. Related Work

Cookie Synchronization: Several studies have been performed on cookie synchronization. Papadopoulos et al. [16] presented a comprehensive study of cookie synchronization in the wild and have found that 97% of Internet users have been exposed to cookie synchronization. They have proposed an approach consisting of blocking all cookies, but this can be problematic as more and more websites will refuse service to anyone using cookies or adblockers. Papadopoulos et al. [17] have shown that if even one unencrypted HTTP request is present in the browsing history, this could potentially compromise the user's full browsing history, even the parts that were sent over HTTPS.

Acar et al. [1] have conducted a study on the collaboration between persistent long-lasting cookies, known as EverCookies, and cookie synchronization. In their study, they have found that several advertising networks on the Internet will respawn previously deleted cookies. This indicates that user awareness is not enough and that any proposed method must work under the assumption that cookies cannot be deleted.

Adobe, in collaboration with PageFair [15], has done an ad blocking report and has found that in 2015, with 16 % of American Internet users using blocking software and a growing percentage of American and European users using ad blocking software, this can cost upwards of 21 billion dollars globally. Since then, the GDPR has come into effect in 2018, and with it a fear from websites that compliance with the GDPR may affect their ad revenue. Therefore, there is an incentive for websites to bypass GPDR compliance [9].

Local Differential Privacy: Differential privacy [5] is a privacy model that provides a rigorous framework for sharing datasets without revealing specific information on any individual in the dataset. Differential privacy guarantees that any computed result from the dataset remains similar regardless of whether the data of a single person is included. Formally, for two databases D_1 and D_2 differing by one data record, a differentially private method creates a database D_2 such that

$$\frac{\Pr[\mathcal{M}(D_1) \in S]}{\Pr[\mathcal{M}(D_2) \in S]} \leq e^{\epsilon} \tag{1}$$

where $\mathcal{M}$ represents the mechanism that takes a dataset as input and produces a differentially private result. ϵ is a user-specified privacy budget, thus giving the user control over how much privacy is required. The lower the value of ϵ, the more similar the information in the database will be if new values are added.

An approach to achieving this is through *local differential privacy*, where users anonymize their personal data before they are incorporated into the dataset. This makes it significantly more difficult for an attacker to infer specific individual's information in the dataset [3]. Dwork [5] presented an early analysis of the feasibility of applying differential privacy in a database. McSherry and Talwar [11] were the first to introduce the concept of the *exponential mechanism* that allows one to modify a value in a database based on a discrete set of results and the utility of each outcome. Niu et al. [13] presented an algorithm called *personalized exponential mechanism*, which allows users to set their own privacy guarantees on the known exponential mechanism.

3. Problem Formulation

The goal of this paper is to mitigate the negative effects of cookie synchronization while continuing to deliver relevant advertisements to users. This challenge can be broken down into two main requirements:

Privacy requirement: This requirement is to limit the privacy risks introduced by cookie synchronization. We considered several syntactic privacy models, such as k-anonymity [23], LKC-privacy [12], etc. However, these models rely on assumptions about the prior knowledge of an adversary and may fail if their respective assumptions are violated. Thus, we decide to enforce local differential privacy [5], which focuses on the semantic impact of the data on the output. A naive way to address this privacy risk would be to prevent all cookies from being stored on the users' browser, but that approach results in websites losing their main source of revenue. Hence, maintaining adequate accuracy in cookie matching is also important, leading us to the second requirement.

Internet usability requirement: Users should be empowered to control their privacy and prevent websites from aggregating data and potentially manipulating their behavior with targeted ads. Users must retain the ability to decide how much information is shared while having unrestricted access to websites across the internet. Third-party ad networks should still have access to user cookies to prevent website restrictions.

We refer to this general challenge as *the problem of privacy-preserving cookie synchronization*, which involves implementing a proxy that satisfies both privacy and internet usability requirements.

4. Cookie Synchronization Proxy

We propose a *cookie anonymization proxy*, which increases user privacy without disrupting normal web browsing. Specifically, our design integrates a proxy into the HTTPS client-server interaction to act as a man-in-the-middle, intercepting, and modifying third-party cookies in transit.

4.1. Hardware Trust and Attested TLS

When a user utilizes a Cookie Synchronization Proxy, the proxy intercepts a significant portion of the user's traffic, thereby gaining potential insight into the user's web activity, including the websites being visited and the content being accessed and retrieved from each of these sites. Web browsers are prevalent applications through which users inadvertently disclose all these details. Nevertheless, the fundamental assumption held by end users is that browsers do not exploit this information, as web browsers are publicly available and scrutinized; if such practices were to be revealed, users would likely abandon and change their browser. In other words, the associated risk is deemed not worth taking. A similar model could be applied to locally installed proxies. However, the maintenance of such services, along with the necessary dynamic adjustments, renders this model challenging to sustain over the long term. Instead, it is anticipated that a third party may offer such services. The primary distinction from the previous deployment model is that the software is managed by a third party, yet the user possesses virtually no insight into which software is actually in use or what actions the operator is undertaking. In essence, with traditional software deployment, the end user would likely place their trust in the service provider, but we believe that this model imposes an excessive level of trust that is untenable. In reality, users can scarcely trust a company that may later be acquired by another entity, subject to various legal and political pressures.

To establish trust in a single company, we examine the Trust Execution Environment (TEE). In this model, the Root of Trust is the hardware (i.e., CPU), which asserts the software that is effectively operational through attestation [20]. Various architectures exist; ultimately, the hardware measures the software being loaded and signs that measurement within a quote. The process of attestation involves verifying the quote, ensuring that the measurement is issued by a trusted entity, specifically the hardware manufacturer, and validating the measurement itself. This measurement characterizes the software. Consequently, attestation guarantees to the user that a specific software is running. In our deployment, we anticipate integrating attestation with the establishment of a TLS session with the Proxy, a process referred to as attested TLS [21]. This ensures that the user can confirm that they are communicating with the Proxy during every TLS session. Maintaining a connection to a particular software is only valuable when there is assurance that the software is performing as anticipated. A viable model for this could involve the open-source release of the software, allowing for the verification of the associated measurements. In our scenario, the publication of the software could guarantee that user information remains within the enclave, is protected from leakage, and is utilized solely for its intended purpose.

4.2. Man-in-the-middle Proxy

When a user visits a website, an algorithm within the proxy determines the odds of the users' cookies for that website being anonymized. Figure 2 illustrates the multistep procedure for two advertising networks, although the concept is generalized to any number of networks and can be applied to multiple users simultaneously. The end result is that Ad Network 2 will have a cookie matching table that indicates that for the website www.website.com, and a cookie of ID *ad2id* set by Ad Network 2, the equivalent cookie set by Ad Network 1 will have an ID of value *anonymizedad1id* – not *ad1id*, the real ID. This means that each Ad Network will maintain different data on each other, and the networks will not be able to synchronize properly.

The purpose of the anonymizing proxy is not to completely prevent synchronization of cookies. Rather, it is meant to add noise to the advertising network's database, so that cookie synchronization is less effective and seen less often as the user browses the Internet. If a user has a long-term identifying cookie placed on their browser, this may identify them during future visits if this cookie is not anonymized, potentially even leading to the ad network going back and syncing an anonymized cookie with the real cookie. However, this risk is mitigated by the fact that the proxy *continuously* anonymizes cookies, meaning that any third party ad network will not be guaranteed to have a complete picture of the user's data. This successfully manages to mitigate the effects of cookie synchronization.

4.3. Applying Differential Privacy

In one of the first examples of local differential privacy, Warner [26] presented a method to ensure user privacy when answering surveys with potentially sensitive information. Suppose a survey wants to collect some sensitive information in a binary set of answers, for example, whether a person belongs to group A or B. They will ask a user if they belong to group A or B, and collect statistics on the total–the potential outcomes are the set $O = \{\mathrm{A}, \mathrm{B}\}$. However, if this is a sensitive question, users may not want to answer truthfully. Thus, a technique was suggested that was proven to preserve the same statistical results: individual privacy could be preserved if that person answered honestly with probability p, or picked an answer in O at random with probability $1 - p$.

Although this technique predates formal differential privacy, it does satisfy differential privacy when dealing with binary values of sensitive information [6]. We applied this method to our proxy. The proxy gives the users two choices when faced with a tracking cookie: keep the cookie as is and potentially lose privacy, or give the ad network a false cookie value that does not match the user. By adjusting the privacy budget in local differential privacy, the user may achieve a balance between the requirements for privacy and data utility. Formally, the set of possible outcomes on cookies would be $O = \{\mathrm{Keep}, \mathrm{Anonymize}\}$, each outcome representing the decision to perform on the value of a cookie.

Although this technique is powerful, a coin flip does not give users much control over the desired level of privacy. This is the reason for a modern differential privacy framework: we wish to allow users to choose the level of privacy they want. In this paper, we have chosen the exponential mechanism [11] as the mechanism $\mathcal{M}$ that will guarantee differential privacy. The exponential mechanism has been used before to anonymize discrete sets of outcomes and uses the user-controlled privacy budget parameter ϵ to guarantee that the probability ratio of observing a particular $O' \in O$ is at most e^{ϵ}. This allows the user direct control over their level of privacy. The exponential mechanism is as follows. The probability of outcome $O' \in O$ is given by [6]:

$$P(x, O') = \frac{\exp\left(\frac{\epsilon \cdot u(x, O')}{2\Delta u}\right)}{\sum_{O' \in O} \exp\left(\frac{\epsilon \cdot u(x, O')}{2\Delta u}\right)} \tag{2}$$

where O' is an outcome $\in O$, x is a data point, ϵ is the privacy budget, Δu is the sensitivity function and $u(x, O')$ is the utility function. The utility function $u(x, O')$ quantifies the desirability of outputting outcome O' with data point x.

To determine the utility of keeping or anonymizing, we use a variation of the Heaviside function, giving each cookie a score $\in \{0, 1\}$ based on whether or not it is desirable for the ad network to keep it. To determine whether a cookie is desired to keep, a heuristic-based filtering system was implemented to identify whether the value of a cookie is an identifier for tracking. Specifically, the system considers that a cookie is *useful* to the advertising network if any of the following three conditions are satisfied: (1) the terms UUID, ID, or user identification are embedded in the value of the cookie, (2) the cookie name is a variation of UUID, ID, or userid, or (3) the cookie value is an alphanumeric string longer than 10 characters [2].

HTTPS transactions typically use a set of multiple cookies in one HTTPS transaction. Let C be the set of cookies in each HTTPS transaction, and $|C|$ be the number of cookies in C. Normally, the exponential mechanism assumes that all data points are independent. However, each cookie in C was set by the same domain in the same user's browser, so they cannot be considered truly independent of each other. For anonymization to be effective, the decision O' must apply to the *entire* set

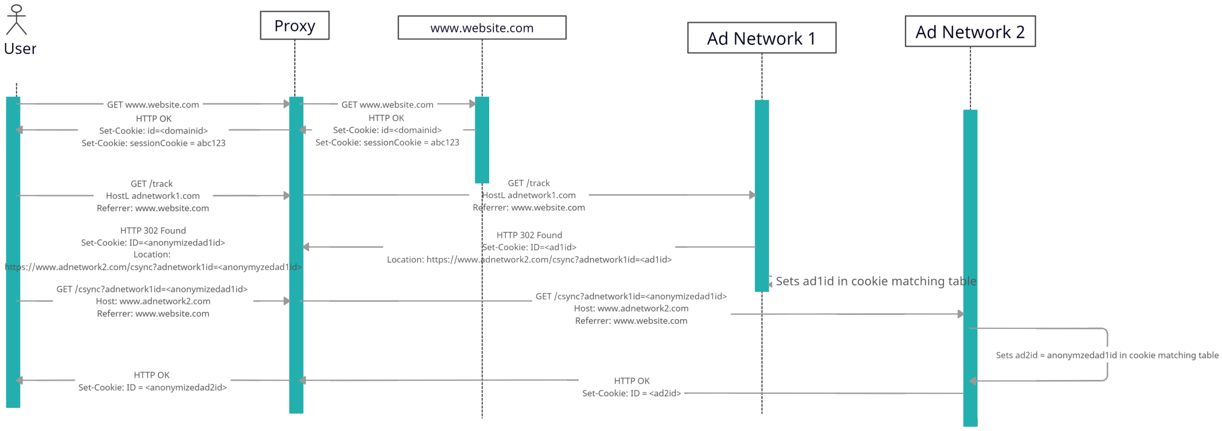
User
Proxy
www.website.com
Ad Network 1
Ad Network 2
GET www.website.com
HTTP OK
Set-Cookie: id=<domainid>
Set-Cookie: sessionCookie = abc123
GET www.website.com
HTTP OK
Set-Cookie: id=<domainid>
Set-Cookie: sessionCookie = abc123
GET /track
HostL adnetwork1.com
Referrer: www.website.com
GET /track
HostL adnetwork1.com
Referrer: www.website.com
HTTP 302 Found
Set-Cookie: ID=<anonymizedad1id>
Location:
https://www.adnetwork2.com/csync?adnetwork1id=<anonymyzedad1id>
HTTP 302 Found
Set-Cookie: ID=<ad1id>
Location: https://www.adnetwork2.com/csync?adnetwork1id=<ad1id>
Sets ad1id in cookie matching table
GET /csync?adnetwork1id=<anonymizedad1id>
Host: www.adnetwork2.com
Referrer: www.website.com
GET /csync?adnetwork1id=<anonymizedad1id>
Host: www.adnetwork2.com
Referrer: www.website.com
Sets ad2id = anonymzedad1id in cookie matching table
HTTP OK
Set-Cookie: ID = <anonymizedad2id>
HTTP OK
Set-Cookie: ID = <ad2id>

Figure 2: Client-ad platform HTTPS flow with *CSyncProxy*

C, or else the advertising network can simply link the cookies based on a common user and resume tracking.

Applying the exponential mechanism to the entire set C has a limitation [10] [14]: the privacy parameter is increased by a magnitude of $|C|$, significantly reducing the mathematical privacy guarantee. However, we get around this limitation by treating the set C as a single *independent data point*, where both $u(C, \text{Anon})$ and $u(C, \text{Keep})$ have values between 0 and $|C|$. In this case, the utility function $u(C, O')$, is calculated as follows, with c_i is cookie in C and $O' \in O$:

$$u(C, \text{Anon}) = \sum_{c_i \in C} \left(\begin{cases} 0, & \text{if } c_i \text{ is a tracking ID;} \\ 1, & \text{otherwise} \end{cases} \right) \tag{3}$$

$$u(C, \text{Keep}) = \sum_{c_i \in C} \left(\begin{cases} 1, & \text{if } c_i \text{ is a tracking ID;} \\ 0, & \text{otherwise} \end{cases} \right) \tag{4}$$

Δu is the sensitivity of the score and measures the maximum change that occurs if one data entry changes. Since we treat set C as one independent data point, the maximum and minimum possible values of $u(C, O')$ are $|C|$ and 0, respectively. We use the formal definition of the sensitivity function Δu as the maximum difference in utility when one data point is changed, in our case the set of cookies C. Therefore, the value of Δu would be $\big||C| - 0\big|$, which would lead to a final value of $|C|$. This value of Δu would apply to both $P(C, \text{Keep})$ and $P(C, \text{Anonymize})$. This is useful because it deals with the concern about decreasing the privacy guarantee by a factor of $|C|$. By dividing $u(C, O')$ by $|C|$, we ensure this algorithm remains ϵ-differentially private.

The anonymization algorithm has the goal of being as close to a real ID as possible. The reason for this is to make it less obvious to a third-party tracker that the cookie value has not been assigned by the tracker but by a proxy. Therefore, we need to take a closer look at the structure of an ID at the character level. If it is a special character, such as a period, a dash, or a slash, we leave it as is. If it is a lowercase letter, an uppercase letter, or a number, we randomly choose another lowercase letter, uppercase letter, or number, respectively, in order to mimic the ID structure of the third-party advertising cookie. For example, if the original value was UUID=A65tyyk776f-09y, then the anonymized value could be UUID=R44tyub453g-12f.

5. Experimental Results

5.1. Dataset and Setting

We have created two different *CSyncProxy* systems[1] to analyze all cookie values exchanged during all HTTPS transactions on the top 100 websites: one that collects cookies without modifying any cookies and another that collects cookies and uses the anonymization process. The former was used to evaluate the effectiveness of the privacy budget ϵ, and then was used as a baseline to compare the performance with the anonymization proxy.

The data used to evaluate the proxy is the list of the *Top Sites of Tranco* [18]. This is a compiled list of the top 10,000 websites per number of visits. This data set is publicly available on the Internet. Crawling the top 100 websites is sufficient because only using the top 100 websites can we still have an accurate view of the synchronization mechanism in nature [8]. The domain, referrer, and cookie value of each HTTPS request and response were collected. In both experiments, 9,000 cookies were collected and analyzed.

This web crawl reflects a lower bound on potential anonymization. In reality, users revisit websites in a Zipf-like distribution [4], frequently returning to top sites. Therefore, the proxy must recognize and re-anonymize cookies from previously visited sites with 100% certainty, without persistently storing user data. This re-anonymization, through methods like tagging cookies, can further improve user privacy in practice.

The environment was a new Google Chrome browser, without plugins, no ad blocking extensions,

[1] https://github.com/McGill-DMaS/CSyncProxy

all cookie blocking settings disabled, and all previous cookies erased. Then a list of potential privacy budget $\epsilon = \{0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 0.75, 1\}$ was tested. These values were chosen to represent a variety of reasonable privacy levels between 0 and 1.

5.2. Performance Metrics

In evaluating proxy performance, we focus on the impact that the proxy has on cookie synchronization. Therefore, we measure the effectiveness of the proxy using *probability of successful anonymization*. The proxy manages to prevent a successful cookie synchronization if the cookies included in both the HTTPS request and response are anonymized. This means that there will be a disconnect between the cookies that the user receives and sends, ensuring a disconnect in the advertising network's cookie matching table. The probability is defined as follows:

$$P(\text{Success}) = P(O' = \text{Anon} \mid u(C, \text{Keep}) > u(C, \text{Anon})) \tag{5}$$

If a cookie value is an ID and is thus useful for ad networks, then $u(C, \text{Keep}) > u(C, \text{Anonymize})$. Thus, the value of $P(\text{Success})$ represents the probability that we anonymize all cookie IDs in an HTTPS transaction, despite the fact that they are useful for the ad network.

There are two types of HTTPS transaction: a request and a response. All that is needed to guarantee successful anonymization is the anonymization of all cookies in *either* in the request or in the response because anonymizing one transaction disrupts the cookie matching process for advertising networks *at that moment*, and with each subsequent transaction there is still a chance of anonymization. The proxy will have prevented one possible instance of cookie synchronization from happening if either a request or a response has been anonymized.

In the first experiment, different values of the privacy budget ϵ were tested to determine the effect of each on $P(\text{Success})$. Then, once the value of ϵ was chosen that ensures the highest value of $P(\text{Success})$, the cookies were anonymized using Equation 2 and the chosen value of ϵ. The number of recorded instances of cookie synchronization was compared between using Chrome with default settings, using Chrome with the *block third-party cookies* setting enabled, and CSyncProxy.

5.3. Analysis

Data collected while anonymizing were analyzed and an average of $P(\text{Success})$ was calculated for each parameter. Figure 3 shows the impact of the privacy budget ϵ on $P(\text{Success})$. In general, $P(\text{Success})$ decreases as ϵ increases. As ϵ approaches 1, the choice to anonymize or not becomes increasingly deterministic, dependent on the value of $u(C, O')$. This means that the proxy will make decisions based on whether the cookie ID is useful to advertising networks. Thus, there is a trend showing that a lower value of ϵ increases the user's privacy. This shows the advantage of multiple values of ϵ, notably by giving users control over their level of privacy.

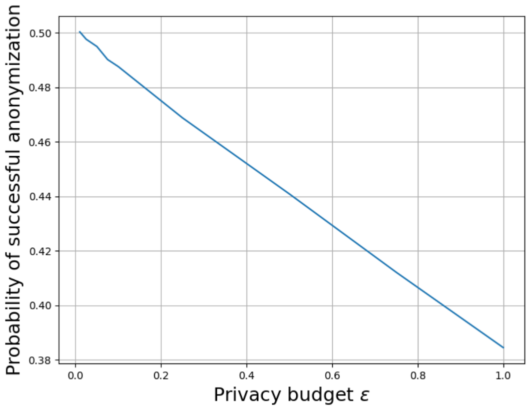

Figure 3: Probability of successful anonymization

Privacy requirements will vary depending on each users behavior. A user who mainly visits low-stakes websites with a lot of tracking cookies–examples being social media or shopping websites–might set a higher value of ϵ (between 0.6 and 1). While privacy isn't their top priority, this still helps them feel less exposed. Someone who visits a combination of low-stakes and high-stakes websites, or who uses the internet more extensively, privacy may be more of a priority and thus ϵ would take a value between 0.4 and 0.6. A user who wants to avoid tracking as much as possible, for example, a user who visits a government website, will prefer a value of ϵ closer to 0. This gives the user much more granular control over their privacy than the simple binary on-off options provided by many adblockers.

We then evaluate the performance of the anonymization proxy in a privacy budget of $\epsilon = 0.01$, which represents the strongest privacy guarantee among the values tested. We ran both the baseline and anonymizing proxy 10 times over the Top 100 websites and took the mean of the recorded instances of cookie synchronization. In the baseline scenario, where cookies are recorded without anonymization, there were **69** instances of cookie synchronization observed on 100 websites. In contrast, with CSyncProxy configured as described above, only **39** instances were recorded, representing a reduction of 43%. This outcome matches the theoretical probability of successful anonymization illustrated in Figure 3 for $\epsilon = 0.01$. A third test was performed with *Block third-party cookies* enabled in the Chrome setting, which yielded an average number of instances of **44**, a 11% increase from CSyncProxy. The *block cookies* option also was more likely to block users from accessing websites. These findings demonstrate that the proxy can reduce the prevalence of cookie synchronization in real-world browsing scenarios by more than 40%.

6. Conclusion and Future Works

A proxy based on differential privacy has been developed to anonymize cookie IDs, reducing the effects of cookie synchronization and limiting large-scale tracking by third-party advertisers. As the first method to apply differential privacy to this problem, it preserves user anonymity while maintaining web usability. It also supports a more balanced approach to cookie data management, aligning the interests of both users and advertising networks.

One limitation of CSyncProxy is added latency: it slows full page loads by 250–300ms on average. However, similar delays occur with just the baseline proxy, while the cookie anonymization logic adds only 65ms. This overhead stems from the proof-of-concept being implemented in Python, which is slower than compiled languages like C++, C# or Rust for network proxy tasks [19] [7]. Future work could explore implementing CSyncProxy at the network level in a compiled language like C++, which would maintain the same privacy guarantees and results, while improving performance, as the anonymization algorithm contributes little to latency.

Further studies could look at applications of the problem of *data brokerage*, which sells large amounts of user data to third parties. The proposed *CSyncProxy* could be used to counteract the effects of data brokerage on user privacy. Further studies could also look at the integration of machine learning concepts into CSyncProxy, specifically with user ID identification, providing a more robust method of catching and identifying potential tracking cookies.

Acknowledgments

This research is supported by MITACS (IT33175), Ericsson, NSERC Discovery Grants (RGPIN-2024-04087), NSERC CREATE Grants (CREATE-554764-2021), and Canada Research Chairs Program (CRC-2019-00041). The authors also acknowledge Hyame Alameddine and Taous Madi, with Ericsson Research Security in Montreal, for reviewing the article.

References

[1] ACAR, G. The web never forgets: Persistent tracking mechanisms in the wild. In *ACM SIGSAC Conference on Computer and Communications Security* (2014), pp. 674–689.

[2] BANGAR, R. Catch me if you can: achieving complete internet anonymity using open source technologie. In *Proceedings of the 7th International Conference on Computing in Engineering Technology (ICCET)* (2022), IET, pp. 1–3.

[3] BEBENSEE, B. Local differential privacy: a tutorial. *CoRR abs/1907.11908* (2019).

[4] BRESLAU, L., CAO, P., FAN, L., PHILLIPS, G., AND SHENKER, S. Web caching and zipf-like distributions: evidence and implications. In *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)* (1999), vol. 1, pp. 126–134 vol.1.

[5] DWORK, C. Differential privacy. In *Automata, Languages and Programming* (Berlin, Germany, 2006), ICALP 2006, pp. 1–12.

[6] DWORK, C., AND ROTH, A. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science 9*, 3–4 (2014), 211–407.

[7] FLUXZY. 30x to 70x faster than mitmproxy/mitmdump, 4x faster than squid, 2022.

[8] GHOSH, A., AND ROTH, A. Selling privacy at auction. *CoRR abs/1011.1375* (2010).

[9] JOHNSON, G. A. Economic research on privacy regulation: Lessons from the gdpr and beyond. In *The Economics of Privacy* (Online, 2022), University of Chicago, pp. 1–40.

[10] MAJEED, A., KHAN, S., AND HWANG, S. Group privacy: An underrated but worth studying research problem in the era of artificial intelligence and big data. *Electronics 11* (04 2022), 1449.

[11] MCSHERRY, F., AND TALWAR, K. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)* (Providence, Rhode Island, 2007), IEEE, pp. 1–10.

[12] MOHAMMED, N., FUNG, B. C. M., HUNG, P. C. K., AND LEE, C. Centralized and distributed anonymization for high-dimensional healthcare data. *ACM Transactions on Knowledge Discovery from Data (TKDD) 4*, 4 (October 2010), 18:1–18:33.

[13] NIU, B. Utility-aware exponential mechanism for personalized differential privacy. In *IEEE Wireless Communications and Networking Conference (WCNC)* (2020), pp. 1–6.

[14] OH, S., AND VISWANATH, P. The composition theorem for differential privacy. *CoRR abs/1311.0776* (2013).

[15] PAGEFAIR, A. The cost of ad blocking, 2015.

[16] PAPADOPOULOS, P. Cookie synchronization: Everything you always wanted to know but were afraid to ask. In *Proceedings of the 2018 World Wide Web Conference* (2018), pp. 1432–1442.

[17] PAPADOPOULOS, P. Exclusive: How the (synced) cookie monster breached my encrypted vpn session. In *Proceedings of the 11th European Workshop on Systems Security* (2018), EuroSec, pp. 1–6.

[18] POCHAT, V. L., GOETHEM, T. V., TAJALIZADEHKHOOB, S., KORCZYŃSKI, M., AND JOOSEN, W. Tranco: A research-oriented top sites ranking hardened against manipulation), 2024.

[19] RETUNSKY, E. Benchmarking low-level i/o: C, c++, rust, golang, java, python, 2021.

[20] SARDAR, M. U., FOSSATI, T., FROST, S., AND XIONG, S. Formal Specification and Verification of Architecturally-defined Attestation Mechanisms in Arm CCA and Intel TDX. *IEEE Access 12* (2024), 361–381.

[21] SARDAR, M. U., NIEMI, A., TSCHOFENIG, H., AND FOSSATI, T. Towards validation of tls 1.3 formal model and vulnerabilities in intel's ra-tls protocol. *IEEE Access 12* (2024), 173670–173685.

[22] STATISTA. Global internet advertising revenue, 2024.

[23] SWEENEY, L. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10*, 05 (2002), 557–570.

[24] UNION, E. General data protection regulation (GDPR), 2018.

[25] W3C. Same origin policy, 2008.

[26] WARNER, S. L. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association 60*, 309 (1965), 63–69. PMID: 12261830.