Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Electrical, Electronics and Communications Engineering
(36^{th}cycle)

Privacy on the Web: Algorithms, Tools and Measurements

By

Nikhil Jha

******

Supervisor(s):
Prof. Marco Mellia, Ph.D., Supervisor
Dr. Martino Trevisan, Ph.D., Co-Supervisor

Politecnico di Torino
2024

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Nikhil Jha
2024

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Marco Mellia: his expertise and guidance have been vital not only as a supervisor, but also as an example of the researcher I would like to become in the future. I have never taken for granted the time he always found for me in the midst of his very busy schedule, and how he always treated me as a colleague, rather than a student. I also want to thank my co-supervisor, Martino Trevisan, and his kindness in explaining me things which, as a Ph.D. candidate, I was already supposed to know. His efficient lightheartedness in day-to-day job inspires me. A thank you to Prof. Pietro Michiardi of EURECOM: I shall never forget how this journey started with a chat in his office. Thank you to the two reviewers of this thesis, Nataliia Bielova and Melek Önen, for their time and precious feedback.

A hearty thanks to my friends and colleagues over these years. Thanks to Elisa, Federico and those I met in EURECOM. Your company helped me when I was far from home, in the sweet town of Antibes. A big thanks also to those in Turin: Andrea, Danilo, Dena, Francesco, Gianluca, Giordano, Luca G., Luca V., Luciano, Kai, Matteo, Philippe, Tailai. You made our office (should I say, our playground?) a place I look forward to get to every day. To Edoardo P., Edoardo S., Maurizio: we share long-lasting memories. To Carola and Francesco, for the undeserved high opinion they have of me. To Stefano, for the days and nights and laughs we had together. A big, big thank you to Andrea: I am now a step closer to finally become your supervisor, although you would deserve a much better one.

To my parents: you paved my way and did the best you could for the journey to be as smooth as possible. I could not be more privileged. Finally, to my partner, companion, soulmate Giulia: we are building a house with strong foundations, and a wonderful window on a bright future. I'll wait for it on a couch with you and Lali, and I am sure it will hold great things.

Abstract

The continuous and ubiquitous collection and exchange of data is at the foundation of the current Web ecosystem. Many of these data may be of sensible kind, being extracted by various means from users' online activity — often with limited awareness of the process by the users themselves. Users' data are useful for a variety of online and offline entities: advertisers can exploit knowledge on users' preferences to show them advertisements tailored on their needs on the webpages they visit, marketers collect users' data to find valuable information for their business activities. One of the most prominent ways to collect users' data are third-parties cookies. They are small text files installed on users' browsers by entities contacted on the Web, containing identifiers that allow third-parties to follow the users along their navigation of the Web.

The push of the industry towards an ever-increasing amount of collected data collides with the right to privacy that should be guaranteed to Web users: collected data can be used to infer private information about the users, by cross-checking obtained data with other, public sources. In the past years, legislators have tried to give users a larger control over the data that are extracted from their use of the Web. This has led to the proliferation of Privacy Banners, that inform the users of the agents and the purposes of the collection.

In this dissertation, we will discuss various aspects on implementing and measuring privacy on the Web: we will start from the role of Privacy Banners in the current Web ecosystem. We will study how users interact with the Banners, and how crawling techniques that aim at taking measures of key metrics in the Web must take into consideration Privacy Banners in order for their empirical estimates to be accurate and close to real-world experience. Moreover, we will discuss the Topics API, a possible solution that goes beyond third-party cookies, in an effort to re-balance the trade-off between data utility and data privacy.

Finally, we will also introduce a study on the privacy properties of z-anonymity, a data anonymization property and algorithm suited for streaming data anonymization. We compare it with the well-known k-anonymity property, and evaluate the utility loss needed to obtain desired levels of privacy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Online tracking on the Web

In the current Web ecosystem, most services monetize the content they offer via online advertising. This has led to a massive, unprecedented collection of personal data, which is essential for Interest-Based Advertising (IBA) and for marketing and business analytics. This scenario created tension between the online industry and users around their privacy.

The collection of personal information often relies on the use of cookies. Cookies are pieces of text stored in a client's browser. Two types of cookies exist: first-party cookies (which are installed by the website that the user is actually visiting) and third-party cookies (which are installed by separate entities that are hosted in the visited website).

By retrieving previously set first-party cookies, a website can recognize the user and improve one's experience, e.g., by remembering the user's language or the preferred theme. However, third-party cookies (and more advanced mechanisms [1, 2]) are used to collect information about users, track them across different websites, and leverage the information for not only personalized ads [3–7].

In the most common scenario, third-party cookies are set by trackers and advertisers present on a great amount of websites. When a user visits the first website of the like, trackers and advertisers install in the client a cookie containing a unique identifier, together with the information that the user has visited such website. When

the user visits a second website, trackers and advertisers can thus retrieve the cookie, and update it with the new visited website. In this way, after few visits, trackers and advertisers are able to reconstruct a profile of the user by observing the visited websites and their topics. The obtained information can be used, for instance, to show tailored advertisements to each user. However, this framework threatens users' privacy [8].

On their hand, public bodies and regulators have started proposing and enforcing regulations to govern the phenomenon. The European Union (EU) was the first to enact a privacy law that applies to a large geographic region. With the 2009 "Cookie Law" directive [9] all websites that use first-party or third-party cookies to track users' behaviour must obtain user consent via a *Privacy Banner* — and must not use cookies the user has refused. In 2018, the introduction of the GDPR [10] brought to more severe penalties against non-compliant websites.

The Privacy Banners represent a *de facto* barrier to experience the Web, both from a human and an automated perspective. Either way, to explore the functionalities offered by the Web it is not possible to ignore the Privacy Banners, as we will extensively discuss in Chapter 3: on one side, users continuously face Privacy Banners, and often accept all the conditions posed by websites (although easier refusal options result in a larger share of rejections, as we discuss in Chapter 3.2). On the other hand, assuming that users tend to accept cookies, we show that is imperative for Web crawlers to take into consideration the Web scenario upon accepting the use of cookies, because the retrieved metrics may change significantly (as we detail in Chapter 3.4).

In this scenario, new IBA techniques are gaining momentum. In particular, Google recently proposed the Topics API to replace third-party cookies as an arguably more private way to provide advertisers valuable information. At the moment of writing Google is planning to deploy the Topics API framework at large on Chrome instances[1]: a thorough, independent analysis of the Topics API privacy guarantees is thus urgent. We provide a first analysis in Chapter 4, describing the behaviour of Topics API and showing that users in the framework suffer risk of re-identification under a threat model that include two or more colluding websites.

[1] https://developer.chrome.com/blog/cookie-countdown-2023oct/, accessed on Monday 22nd January, 2024.

1.2 Privacy-Preserving Data Publishing

Online advertising is not the only purpose for which data can be collected on and beyond the Web. Big data have opened new opportunities to collect, store, process and, most of all, monetize data. This has created tension with privacy, especially regarding information about individuals.

Anonymization, i.e., generalizing or removing identifying data of individuals, is the classical approach to publish personal information. Thanks to this, Privacy-Preserving Data Publishing (PPDP) has gained attention in the last decade [11].

In Chapter 5 of this thesis, we study the novel anonymization property called z-anonymity, or z-anon for short, previously introduced in the context of Internet traffic analysis. z-anon specifically targets the data stream scenario, aiming to guarantee zero-delay release of data (hence the z). We suppose to receive a raw stream of data in which users' attributes (e.g., visited websites) arrive as they are generated. These attributes are QIs, and may allow users' re-identification. To prevent this, a new attribute is released only if it was exposed by at least $z-1$ other individuals in the past window Δt.

Given the z-anon algorithm, we aim at understanding the privacy guarantees that it offers. To do so, we compare its properties against those of k-anonymity, a well-known static data anonymity property [12]. We use k-anonymity as a benchmark because it is the privacy paradigm on which z-anon itself is based. We test how the values of z relate to the probability of a user being being k-anonymized, under different parameter settings.

1.3 Thesis outline

To sum up, the rest of the present thesis is organized as follows: in Chapter 2, we present all the background work and knowledge concerning the topics of the thesis. In Chapter 3, we will discuss the role that Privacy Banners have both in human and crawling interaction with websites This chapter is mostly based on two works: the first was presented at the *2023 TMA Conference* [13], and the second published on *ACM Transactions on the Web* [14]. In Chapter 4, we will discuss Google's Topics API privacy guarantees (extending the work presented at the *2023 Privacy*

Enhancing Symposium [15]), while in Chapter 5 we will study the privacy properties of the z-anonymity streaming anonymization algorithm, and how it relates to the well-established k-anonymity property. We drew the chapter mostly from two works on the topics, the first presented at the 2020 IEEE International Conference on Big Data (BigData) [16] and the second published on *Performance Evaluation* [17]. Finally, in Chapter 6 we will draw the final conclusions.

1.4 List of publications

The following list encompasses all the publications published as a Ph.D. candidate:

- **z-anonymity: Zero-Delay Anonymization for Data Streams**. Jha, Nikhil; Favale, Thomas; Vassio, Luca; Trevisan, Martino; Mellia, Marco (2021). In: 2020 IEEE International Conference on Big Data (Big Data) [16].
- **A PIMS Development Kit for New Personal Data Platforms**. Jha, Nikhil; Trevisan, Martino; Vassio, Luca; Mellia, Marco; Traverso, Stefano; Garcia-Recuero, Alvaro; Laoutaris, Nikolaos; Mehrjoo, Amir; Azcoitia, Santiago Andres; Rumin, Ruben Cuevas; Katevas, Kleomenis; Papadopoulos, Panagiotis; Kourtellis, Nicolas; Gonzalez, Roberto; Olivares, Xavi; Kalatzantonakis-Jullien, George-Marios (2022). In: IEEE Internet Computing [18].
- **The Internet with Privacy Policies: Measuring The Web Upon Consent**. Jha, Nikhil; Trevisan, Martino; Vassio, Luca; Mellia, Marco (2022). In: ACM Transactions on the Web [14].
- **Practical anonymization for data streams: z-anonymity and relation with k-anonymity**. Jha, Nikhil; Vassio, Luca; Trevisan, Martino; Leonardi, Emilio; Mellia, Marco (2023). In: Performance Evaluation [17].
- **I Refuse if You Let Me: Studying User Behavior with Privacy Banners at Scale**. ha, Nikhil; Trevisan, Martino; Mellia, Marco; Irarrazaval, Rodrigo; Fernandez, Daniel (2023). In: 7th Network Traffic Measurement and Analysis Conference (TMA). [13]

- **On the Robustness of Topics API to a Re-Identification Attack**. Jha, Nikhil; Trevisan, Martino; Leonardi, Emilio;Mellia, Marco (2023). In: 23rd Privacy Enhancing Technologies Symposium (PETS2023) [15].

Chapter 2

Background

In this chapter, we will discuss the relevant related work for the thesis. The chapter is organized as follows: in Chapter 2.1, we will present the past work in the field of the use of tracking techniques on the Web, the role of the legislators and existing tools that face the pervasiveness of Consent Banners on the Web. In Chapter 2.2, we will introduce the interest-based advertising techniques that preceded the development of Topics API. Finally, in Chapter 2.3, we will discuss related work in the context of privacy-preserving stream data publishing.

2.1 Online tracking

Content providers on the Web often monetize the content they offer by using advertisements. To increase their effectiveness, the so-called interest-based advertisement (IBA) leverages users' interests to provide targeted ads. This is possible thanks to Web trackers, i.e., third-party services embedded in the webpages that gather users' browsing history. Trackers are nowadays largely present on websites and reach the majority of Web users [19, 20]. Trackers exploit cookies and advanced techniques to enable the collection of personal information [21–23].

When a user visits a website for the first time, they have to interact with the Privacy Banner, and, only after getting the explicit user's consent, the website (and any third party embedded in the website) could install cookies and start the data collection. Privacy Banners, however, do not fully protect users in many cases [24].

Fig. 2.1 Example of Privacy Banner on `dailymail.co.uk`. Only upon consent, trackers are contacted and ads displayed.

2.1.1 The role of legislators

In this tangled picture, legislators started to regulate the ecosystem to avoid massive indiscriminate tracking that may threaten users' privacy. In 2013, the introduction of the European Cookie Law [9] mandated websites to ask for informed consent before using any profiling technology. This led to the proliferation of Privacy Banners [25].

Later, in May 2018, the General Data Protection Regulation (GDPR) [10] entered into force in all European member states. GDPR is an extensive regulation on privacy, aiming at protecting users' privacy by imposing strict rules when handling personal information. Unlike previous regulations, it sets severe fines and infringements that could result in a fine of up to €10 million, or 2% of the firm's worldwide annual revenue, whichever amount is higher. Some websites have already been caught to present legal violations in their Privacy Banner implementation [26] and a large fraction have been shown to use tracking technologies before user consent [24, 27]. In the US, the California Consumer Privacy Act (CCPA) [28] enhances privacy rights and consumer protection for California residents by requiring businesses to give consumers notices about their privacy practices.

2.1.2 The human interaction with Privacy Banners

The GDPR has profoundly influenced the Internet user experience [27, 29–32], at least for EU-based users, also defining severe sanctions for violators. Most websites base their business model on advertising, which, in turn, requires that users accept the use of cookies and the collection of personal data. Thus, some websites and CMPs

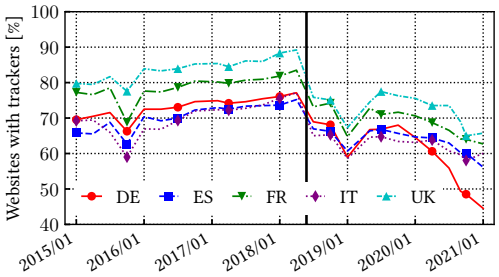


Fig. 2.2 Percentage of websites containing at least one tracker for five European Top-Level domains (from HTTPArchive). The black vertical line indicates the entry into force of the GDPR.

make efforts to increase the cookie acceptance rate. As a result, most of the websites now provide explicit Privacy Banners [33] and many adopt Consent Management toolsets [34], making the website content difficult to access until visitors accept the privacy policy. For example, Figure 2.1 shows the same news website homepage before and after accepting the privacy policy. Only upon pressing the "Got it" button, the website content is fully loaded and visible.

Recent research has shown that banners often nudge users to acceptance by exploiting dark patterns in the user interface, if not openly disregarding GDPR's requirements [26], or making it difficult for users to exercise their rights [35]. They also hinder automated web measurement, hiding the true content of a website, which is visible only upon cookie acceptance [14].

Nudging includes offering the user a `Accept All` default button via intrusive banners [36, 37], which is often the case [38] with websites presenting large pop-ups or wall-style banners that cover most of the webpage content. Researches have shown that apparently minor design choices have a significant effect on inducing the user to accept the cookies [39–44].

In general, it has been shown that most users tend to ignore privacy-related notices [45–47], up to getting annoyed by these. This behaviour has gone under the name of "privacy paradox": users claim to be concerned about their privacy, while at the same time taking little actions to protect their data [48].

2.1.3 The effect of Privacy Banners on Web measurements

Despite cases of misuse, the new regulations had a large impact on the web users and complicate the measurement of the tracking ecosystem. A simple Web crawler visiting the websites without accepting the privacy policies would offer a biased picture, with no tracker and no ad being loaded. Hu *et al.* [49] already found that the number of third-parties dropped by more than 10% after GDPR when visiting websites automatically. Conversely, when using a dataset from 15 real users, they measure no significant reduction in long-term numbers of third-party cookies. Dabrowski *et al.* [29] draw similar conclusions, finding an apparent decrease in the use of persistent cookies from 2016 to 2018. Sorensen *et al.* [50] testify a decreasing trend in the number of third parties during 2018. I quantify this phenomenon in Figure 2.2, using the HTTPArchive open dataset [51]. The curators of this dataset maintain a list of top websites worldwide that they automatically visit using the Google Chrome browser from a US-based server to store a copy of each visited webpage. Using the tracker list detailed in Chapter 3.3, we report the percentage of websites embedding one or more trackers for 5 European countries (simply using the Top-Level Domain to identify the country).[1] We restrict the analysis on those websites that exist for the whole six years-long periods (9 196 website in total).

Figure 2.2 could suggest that the introduction of the GDPR (the black vertical line in May 2018) results in an abrupt decrease in the number of tracker-embedding websites, a trend that continues up to the moment we write. However, as we will show, these measurements are an artifact due to the GDPR itself. Indeed, the Web crawler used by HTTPArchive can only capture the behavior of the websites as a "first-time visitor", before the user accepts any privacy policy. The crawler thus misses third-party trackers and ads.

Research papers that rely on crawling large portions of the Web for different reasons could be affected by the same bias in their measurements. For instance, this would challenge the automatic measurement of the Web ecosystem on privacy [21, 52, 19, 20, 53, 54, 49, 22, 55, 23, 56] and counter-measurements [20, 57, 58]. Moreover, this will also impact those works that rely on crawlers and headless browsers [59] to quantify the impact in the wild of new technologies like SPDY, HTTP/2 [60–63], 4G/5G [64, 65], accelerating proxies [66–68], or generic benchmark solutions [69].

[1] The Top-Level Domain can sometimes be an inaccurate proxy for a website's country. Here, our goal is only to provide a qualitative picture.

At last, even spiders and mirroring tools like Wayback Machine and HTTPArchive may be affected if the website allows the visitor to access its content only after accepting the privacy policy.

2.1.4 Tools managing Privacy Banners

Vallina *et al.* [70] are the first to consider the impact of the Privacy Banner presence. First, they instruct a custom OpenWPM crawler to identify specific Privacy Banners, and then they manually verify the results. Unfortunately, they solely focus on the pornographic ecosystem, which they acknowledge to be rather different from the Web at large, and thus their work can hardly generalize.

Recently, authors of [56] demonstrated that it is fundamental to consider the complexity of the Web ecosystem and include internal pages in every measurement study. They find a number of recent works that neglect internal pages and, as such, might provide biased results. Yet, they ignore the complications due to Privacy Banners. In Chapter 3, we aim at providing an extensive and thorough study of their impact on the Web. Our goal is to enable the automatic study of webpage characteristics as visitors would experience, assuming that most of them accept the default privacy setting as offered by the Privacy Banner. Indeed, it has been shown that most users tend to ignore privacy-related notices [45–47]. Considering GDPR Privacy Banners, users tend to accept privacy policies when offered a default button via intrusive banners that nudge users [36, 37], which is often the case [38] with websites presenting large pop-ups or wall-style banners that cover most of the webpage as seen in Figure 2.1.

For completeness, notice that cookies are among the simplest tracking mechanisms. Authors of [23] show how practices like cookie synchronization, cookie leaking, and other profiling techniques like canvas fingerprinting are common in today's Web. Similarly, authors of [71] show how the crawling context, in terms of vantage point and browser configuration, has a significant impact on the results. Our work is orthogonal to these to obtain automatic, realistic, reliable and user-centric measurements of the Web.

Focusing on automatic management of Privacy Banners, some browser add-ons try to hide them by using a list of CSS selectors of known Privacy Banners. The most popular add-ons of this kind are "I don't care about cookies" [72] and

"Remove Cookie Banners" [73]. Unfortunately, hiding the Privacy Banners has an unpredictable behavior, in some cases falling back to privacy policies acceptance, while, in other cases, triggering an opt-out choice. Other proposals, again in the form of browser add-ons, try to explicitly opt-in or opt-out to cookies. For example, "Ninja Cookie" [74] approves only cookies strictly needed to proceed on the website. Conversely, Autoconsent [75] and Consent-O-Matic [76] use a set of predefined rules to either opt-in or opt-out to cookies, according to the user configuration.

2.2 The Topics API

The implications of web tracking on users' privacy have become more and more debated by the industry [77] and by the research community [78, 4, 79].

Federated Learning of Cohorts (FLoC) has been the first public effort by Google to go beyond the classical web tracking based on third-party cookies [80]. In FLoC, users were grouped in cohorts according to the interests inferred by each one's browser. When asking for information about a user visiting a website, third parties were offered the user's cohort, from which they could have information about the user's interests. In the intention of the proposal, FLoC provided an acceptable utility for the advertisers, while hiding the users (and thus, their identity) behind a group of peers [81]. However, criticism arose around the easiness for first- and third-party cookies to follow the user over time exploiting the sequence of cohorts to which she belongs to isolate and thus identify her [82]. The attack can exploit browser fingerprint to further improve its effectiveness [83]. FLoC's privacy anonymity properties can be broken in several ways [84]. As a response to the critics towards FLoC, Google retired the proposal and conceived the Topics API, whose functioning we describe in Chapter 4.

The Topics API exposes users' profiles in terms of topics of interest to the websites and advertising platforms.Past works already demonstrated that profiling users based on their browsing activity can present severe risks to the privacy of the users [79]. They can be identified with high probability based on the sequence of visited websites [85–87]. Mitigation such as partitioned storage has been put in place to limit the risk, but ways to bypass them exist [88].

Specifically to the Topics API, the same threat we analyze has been already identified by Epasto *et al.* [89] from Google. The authors carry out an information theory analysis and conclude that the attack is hardly feasible.Thomson [90] from Mozilla issued the first independent study on the privacy guarantees of Topics API, elaborating on the conclusions by Epasto *et al.* [89]. He, again, used analytical models and raised severe concerns about the offered privacy guarantees. Recently, Carey *et al.* [91] from Google discuss the privacy implications of the Topics API. They define a theoretical framework to determine re-identification risk and propose the Asymmetric Weighted Hamming Attack (AWHA) for re-identification.

2.3 Data anonymization

On the Web and beyond, the increasing attention to Privacy-Preserving Data Publishing (PPDP) has led to the problem of providing anonymization guarantees to streaming datasets. Most current solutions work with the concept of data batches, i.e., the incoming data are first accumulated, then processed, and finally released with a sizeable delay. Researchers have proposed several approaches during the years that we summarize in the following.

2.3.1 Anonymizing static data

Anonymization, i.e., generalizing or removing identifying data of individuals, is the classical approach to publish personal information. Thanks to this, Privacy-Preserving Data Publishing (PPDP) has gained attention in the last decade [11].

Removing the users' *identifiers* (name, social security number, phone number, etc.) is insufficient to make a dataset anonymous. Indeed, an attacker can link users' apparently harmless attributes (such as gender, zip code, date of birth, etc.) called *quasi-identifiers* (QIs) to some external knowledge. With that, the attacker can re-identify the person and thus gain access to other sensible information available in the dataset (disease, income, etc.) – called *sensitive attributes* (SAs) [92]. Famous is the de-anonymization of the Netflix public dataset [93] based on the exploitation of QIs.

Researchers have defined several properties that data should satisfy to avoid re-identification, among which the k-anonymity [12], or k-anon for short, has become popular. It imposes that every user's released piece of information (a record) should correspond to at least $k-1$ other users, i.e., there are at least $k-1$ other users with the same record. k-anon is conceived for tabular and static data; in other words, the dataset must be completely available at anonymization time.

2.3.2 Anonymizing streaming data

Researchers have proposed extensions of k-anonymity to a streaming scenario, where continuously incoming records are accumulated, processed in batches, and released after an unavoidable delay [94]. When working with batches or microbatches, popular approaches aim at guaranteeing k-anonymity independently in each batch. Here, popular solutions are based on trees [95–97] or clustering [98, 99, 94, 100, 101]. The rationale behind them is roughly the same: firstly load the incoming records into a structure and secondly release tuples for which k-anon is achieved. For instance, authors of [102] design a solution in four steps: cluster arriving tuples, evaluate a noisy centroid for each cluster, control the cluster size to manage concept drifts, and finally release the tuples for those clusters where k-anon is verified. Delay is inevitable with clusters needing to accumulate more data before the release. Authors of [103] modify the attributes to steer a microaggregation process. Tuples are not published as is, but they are first aggregated into clusters, whose only centroid is published. The proposal in [104] uses instead a sliding window approach, where tuples are processed to achieve anonymization using a noisy function. Again, data is released only after a window of time.

Other works design approaches based on perturbation. Authors of [105] propose to perturb the output stream as follows. When a user exposes a sensitive attribute, the system publishes $l-1$ different sensitive attributes so that the attacker can find the actual one with a probability $1/l$. Authors of [106] also propose to replace incoming tuples with sensitive-value sets. To build appropriate sets, they introduce the concepts of semantic and sensitivity diversity. These two techniques allow zero-delay anonymization, but the output streams are largely modified by the perturbation, creating scalability issues too. Furthermore, no guarantees are provided that the resulting release is k-anonymized.

Considering suppression and generalization, authors of [107] propose two algorithms to avoid a correlation analysis from transaction items. They use a sliding window approach and aim to provide the data in the current window as output, after guaranteeing it meets privacy constraints. Again, data is published as a continuous stream, but only after one delay window. At last, researchers have spent some efforts to reduce and factor the cost of the delay: Authors of [96] include the delay in the concept of output quality, with a trade-off between data quality and batch size.

To the best of our knowledge, however, only [105] and [106] target the zero-delay goal. In [105], input streams are anonymized immediately with counterfeit values. In [106], $m-1$ random sensitive attributes are published with the original one.

Chapter 3

The impact of Privacy Banners on the Web

Privacy banners are are an ever-present feature of one's experience of the Web. Due to the introduction of several regulations [9, 10, 28], websites have been forced to show the users that visit them a Privacy Banner: by interacting with it, users can detail their preferences regarding the use of cookies during their visit on the website.

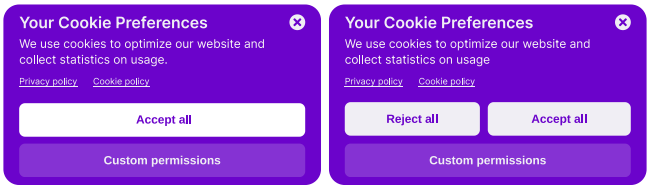
The current chapter is divided in two part: in the first one (Chapter 3.1 and Chapter 3.2), we present how users interact with Privacy Banners and how the design choices of the banners influence users' behaviours. In the second part (Chapter 3.3, Chapter 3.4, and Chapter 3.5), we observe how ignoring the Privacy Banners may cause a distorted understanding of the Web features.

The current chapter is mostly taken from two published articles: the first one presented at the *2023 TMA Conference* [13], the second one published on *ACM Transactions on the Web* [14].

3.1 A Consent Management Platform's data

3.1.1 The Privacy Banner

For what concerns users' interaction with Privacy Banners, we rely on data collected within a medium-sized Consent Management Platform (CMP). A CMP provides



(a) Default banner without the `Reject All` button.

(b) Banner with the `Reject All` button.

Fig. 3.1 Privacy Banners presented to users.

Web developers with the ability to install a simple Privacy Banner to enable/disable data collection via cookies or other advanced means. In our case, the banner takes the form of a small overlay window that can be placed in different parts of the screen. We show it in Figure 3.1. The shape is the same on both desktop and mobile devices. The user is offered an `Accept All` button to accept all cookies at once and a `Custom Permissions` button (Figure 3.1a). The latter brings the user to a second window where they can select which cookies to accept from a short list of categories. These include (i) necessary , (ii) statistical, (iii) preferential, and (iv) marketing cookies. Necessary cookies cannot be deactivated as they are vital for the website operation. Depending on the website, the Privacy Banner is shown on the top or on the bottom of the webpage. In the latter case, the website administrator can choose to show it as a rectangle (default behavior, as in Figure 3.1a) or in a square shape in the bottom-left corner of the screen. At last, the banner offers direct links to the website cookie and privacy policy. Both policies contain details about which data the site collects and for what purposes, and which cookies the system uses, including third-party ones.

The `Reject All` button

The latest practices regarding cookie management in GDPR countries recommend the Privacy Banners to offer a `Reject All` button. This is a consequence of the fine imposed by CNIL (the French data protection authority) on Google and Facebook in January 2022. [1] The two companies were fined for using confusing language in their Privacy Banners, and for making it difficult to opt out of cookie usage. In fact, it was not as easy to reject cookies as it was to accept them, and this was considered a form of dark pattern that nudges users to provide their consent. Since the last week

[1] https://www.cnil.fr/en/cookies-cnil-fines-google-total-150-million-euros-and-facebook-60-million-euros-non-compliance – accessed on Monday 22nd January, 2024

of August 2022, the CMP analyzed in this study has updated its solution to offer a `Reject All` button (Figure 3.1b). If selected, the system will disable all cookies except the necessary ones. The button bears the text `Reject All` and has a similar shape and style as the `Custom Permissions` button. This button is only shown to visitors of GDPR countries after August 25, 2022.[2]

3.1.2 Data Collection

The CMP collects data when users interact with the Privacy Banner shown on a website. The collection happens when the user submits their preference. No data is collected if users do not perform any action on the banner. In details, after the user's selection, the CMP sets a necessary cookie on user's browser to store their preference, and logs data about the time of the visit, the website showing the banner, which cookies the user accepted, the user agent offered by the browser, and the user country of origin, obtained through the client IP /24 subnet geolocation via the MaxMind GeoIP[3] database.[4] This information is necessary to implement the functionalities of the platform (i.e., record user's preferences for the next visits to the website), and it is useful to customize the information provided to users (e.g., show the banner in different languages, show the `Reject All` button if needed), to collect statistics about the usage of the platform, and to bill the website deploying the CMP. All these pieces of information are documented in the privacy policy the CMP offers to users.

Each user who submits (or changes) their preferences generates an entry in the log, which we call *interaction*. Each entry is associated with a random user-id. This makes it impossible to re-identify or track a user across different websites, guaranteeing user's privacy. To further protect the customers of the CMP, the website name is also anonymized by replacing the domain name with a random identifier.

Ethical Aspects In this study, we adopted a lawful and ethical methodology for data collection and processing. First of all, users who interact with the Privacy Banner must accept technical cookies and thus the privacy policy. Indeed, technical cookies

[2]"GDPR countries" refers to any European country where the GDPR is in place. This includes U.K. which adopted GDPR in the "Data Protection Act" in 2018.

[3]https://www.maxmind.com/

[4]We do not consider IPv6, as it generates negligible traffic.

Table 3.1 Summary of the two periods we use to compare user behavior on Privacy Banners with or without the `Reject All` button.

Period	**Start**	**End**	`Reject All` **button**
Period A	Jul 1, 2022	Aug 24, 2022	Not present
Period B	Aug 25, 2022	Oct 4, 2022	Only for users from GDPR countries

Table 3.2 Number of interactions per geographical region.

Region	**# of interactions**	**% of interactions**
Latin America	3 750 135	93.28%
North America	153 365	3.81%
GDPR-regulated	71 640	1.78%
Africa	31 917	0.79%
Asia	7 722	0.19%
Oceania	2 782	0.07%
Rest of Europe	2 691	0.07%

are mandatory to store the user's choice. As said, the Privacy Banners explicitly list "carrying out statistics, managing incidents or conducting market studies" as one of the data collection purposes. Our work fits this purpose. Conversely, we do not collect any data for those users who did not accept technical cookies and thus the privacy policy. Second, we argue that the data we collect can hardly be considered "personal data". we only collect the $/24$ subnet and the user agent to extract user's country and device. Neither the $/24$ subnet nor the user agent are personal data and do not carry information relating to an identified or identifiable natural person.

3.1.3 Data Pre-Processing

We conduct our analysis from the 1st of July 2022 to the beginning of October 2022. In total, we observe 4 million interactions generated by users that interacted with the CMP banner at least once on the 434 websites recorded during the three-month measurement period. Most visitors (93%) are located in South America (where the main business of the CMP is). The remaining ones come from other continents, and we breakdown the audience provenience in Table 3.2. We consider and properly

address this unbalance in the data for our upcoming analysis to provide results which are not biased by the unequal distribution of countries.

Websites belong to different categories, including e-commerce portals and educational institutions. Globally, the CMP manages between 20 k to 30 k new interactions on a daily basis – i.e., new users that come across the CMP Privacy Banner and interact with it.

For each interaction, we compute the choice the user performed according to the combination of accepted cookie categories. In details, we classify interactions as:

- *Accepted-All*: if all cookie categories were accepted, either with a single click on the `Accept All` button, or by individually accepting all the cookies after clicking on `Custom Permissions` button;
- *Mandatory-Only*: if only the necessary cookies were accepted, either by clicking `Reject All` button if present, or by manually deactivating all the cookies after clicking on `Custom Permissions` (with the exception of necessary cookies);
- *Custom*: if at least one among the optional statistical, preferential and marketing cookies was accepted through the `Custom Permissions` screen.

For simplicity, we introduce the class *Reject-Some* to indicate the union of *Mandatory-Only* and *Custom*. These include all interactions but *Accepted-All* – i.e., those in which the user did not accepted all cookies.

To analyze the impact of the presence of the `Reject All` button, we define two measurement periods as detailed in Table 3.1. The first period extends from the beginning of July to August 24, 2022. During this period, the Privacy Banner only included the `Accept All` and `Custom Permissions` buttons as shown in Figure 3.1a. We call it *Period A*. The second period starts on August 25, 2022 and ends on October 4, 2022. Here, visitors from any GDPR-regulated countries face the new version of the Privacy Banner with the additional `Reject All` button, sketched in Figure 3.1b. We refer to this period as *Period B*. We use these two periods to contrast user's behavior with different options in the Privacy Banner. In particular, our dataset allows us to measure the extent to which users reject cookies when the banner provides an immediate opportunity to do so (or not).

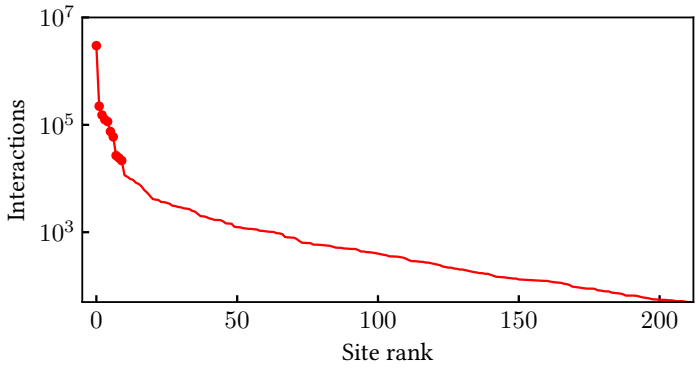


Fig. 3.2 The number of interactions for each website. The markers indicate the values for the top-10 websites.

3.1.4 Dataset Analysis

We now briefly describe the dataset and detail the analysis methodology we design to avoid possible bias in the study. The CMP is present on 434 websites that have a very different audience. Some of them are very popular and generate more than 1 M interactions in total. To characterize the website popularity, we show the volumes of interactions per website in Figure 3.2. Sites are sorted in decreasing number of interactions (notice the log scale on the y-axis). We observe that top websites receive most of the interactions. We record 222 websites collecting less than 50 interactions.

Given the large imbalance in the website audience, we want to prevent large websites from biasing the results. For this, we opt to show results using a website-wise macro-average of the metrics under study. In other words, we compute the desired metric separately for each website. Then we compute the average over the websites. In such way, each website has the same weight in the final metric, regardless of the number of interactions it received.

Formally, given a target metric M, a set of websites $\mathscr{W}$, a population of interactions on a website $\mathscr{I}_w$, a function $\mathscr{M}(M,i)$ which return 1 if i refers to M, 0 otherwise (e.g., whether interaction i records a *Reject-Some* choice or not), we define as $\overline{M}(\mathscr{I})$ the website-wise macro-average of M computed over the samples belonging to the subset $\mathscr{I} = \bigcup_{w \in W} \mathscr{I}_w$:

$$\overline{M}(\mathscr{I}) = \frac{1}{|\mathscr{W}|} \sum_{w \in \mathscr{W}} \left[\frac{1}{|\mathscr{I}_w|} \sum_{i \in \mathscr{I}_w} (\mathscr{M}(M,i)) \right]. \tag{3.1}$$

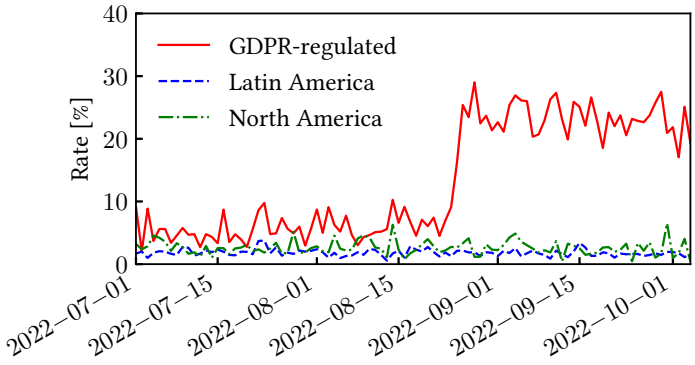


Fig. 3.3 Temporal evolution of the per-region *Reject-Some* rate.

Together with the macro-average, we also evaluate a confidence interval of such average. Hence, each estimate is presented as:

$$\overline{M}(\mathscr{I}) \pm c \cdot \frac{\overline{S}(\mathscr{I})}{|W|},$$

where c corresponds to the quantile of a Student's t-distribution with $|W|-1$ degrees of freedom, and $\overline{S}(\mathscr{I})$ is the sample standard deviation of each website-wise average. In this work, we consider a confidence interval of 90% and report the confidence interval as an error bar. As our main target metric we consider the *Reject-Some* rate.

3.2 Users and Privacy Banners

We now first dissect user behaviour by geographic region and show the impact of adding the `Reject All` button in GDPR countries. Next, we investigate the role of other factors, such as user device and privacy banner position. Finally, we examine the behaviour of users who have particular interactions with the Privacy Banners, i.e., custom choices (*Custom* interactions) or access to the website privacy policy.

3.2.1 Region-wise temporal analysis

We first show the evolution of the *Reject-Some* rate over time in Figure 3.3, separately for the three most represented geographic regions in the dataset. Here, for each day, we compute the *Reject-Some* rate for each website (and region) and then average the values to obtain the macro average. Notice that it sums both the *Mandatory-Only* and *Custom* rates. To avoid websites with very few interactions affecting the results, we evaluated the per-day average only on the websites recording at least 10 interactions on that day. We first observe that the rate exhibits a flat trend for North and Latin America and settles to values in the order of 2%. In European countries where GDPR is in force (solid red line), the *Reject-Some* rate is in the order of 3.5% until August 24 and then jumps over 20%. This increase corresponds to the transition between *Period A* and *Period B* and provides a first quantification of the impact of the `Reject All` button. In the following, we will analyze this in depth.

Notice that new websites have become CMP customers during the observation period (while few have left the CMP). The trend has been increasing over the months as many new websites have been more numerous than desertions. While on the first weeks of July 2022, we find approximately 50 websites every day with more than 50 interactions, on the first week of October 2022, this number increases to ≈ 120. Finally, we observe that the *Accepted-All*, *Custom* and *Mandatory-Only* rates do not depend on the website popularity. If we compute a linear regression using rank as the independent variable and the rates as dependent variables, we obtain a first-order regression coefficient very close to 0. Thus, we can exclude that website popularity plays a role in how users interact with the Privacy Banner.

3.2.2 Geographic Region and `Reject All`

We compare the behaviour of users in different regions of the world. As described in Chapter 3.1.3, the CMP implements a Privacy Banner that can take two forms, during *Period A* and *Period B*.

In Figure 3.4 we provide a breakdown by different geographic regions of the world for the two periods. We group countries by continent but partition Europe in two subsets, considering i) the countries that are part of the European Union (EU) where the General Data Protection Regulation (GDPR) is in force, and ii) all the

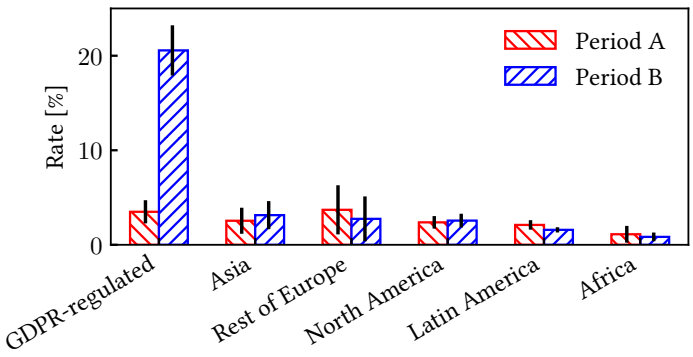


Fig. 3.4 Users' *Reject-Some* rate in different continents, before and after the introduction of `Reject All` button in GDPR countries.

others. We consider the United Kingdom a GDPR-compliant country because it has a nearly identical regulation. To ensure a fair comparison, we show only the regions for which at least 10 websites had 10 interactions or more in both *Period A* and *Period B*. The red bars show the *Reject-Some* rate during *Period A*; and the blue bars during *Period B*. As described in Chapter 3.1.4, the values of the bars represent the website-wise macro-average of the rate. Thus, each website has the same weight. The vertical black lines indicate the 90% confidence interval for such average.

Starting from *Period A* (red bars), we do not observe significant differences between regions when all users are shown the Privacy Banner as in Figure 3.1a, as confidence interval bars overlap, with the exception of GDPR-regulated countries and African countries. In all cases, the rate is primarily due to *Mandatory-Only* interactions, while the percentage of *Custom* interactions is negligible (on the order of 0.1-0.2%).

The blue bars in Figure 3.4 report the *Reject-Some* rate during *Period B* when users from GDPR countries see the Privacy Banner with the additional `Reject All` button as in Figure 3.1b. In these countries, grouped on the left of the figure, we observe a sharp increase by a factor of four. The *Reject-Some* rate grows from 3.49% to 20.56%. Non-overlapping error bars show this increase is statistically significant. As expected, we do not observe any significant changes for the other geographic regions as users still interact with the first version of the banner. Overall, this figure shows how the design of the Privacy Banner influences users' decisions. When it is as easy to reject cookies as it is to accept them, more than one in five users chooses

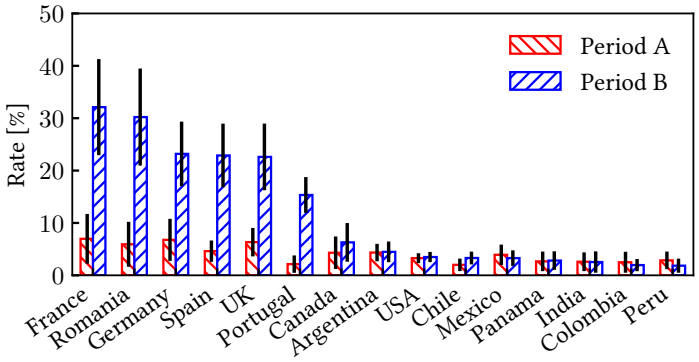


Fig. 3.5 *Reject-Some* rates according to user's country, sorted in descending order by the rate in *Period B*.

to reject them. As a consequence of CNIL fines on Google and Facebook, many European websites and CMPs are implementing similar `Reject All` buttons. In general, we can relate these results to the debate about dark patterns [26, 39]. Our measurements confirm how the options present in the Privacy Banner can influence users' choices on cookies and reveal a nearly 5× increase in users rejecting cookies when only a single click is required. We stress the importance of being able to quantitatively evaluate said figures. It is interesting to observe that the large fraction of users who opt out of cookies with such a Privacy Banner can somehow impact the business of those portals that rely heavily on tracking and behavioural advertising.

Figure 3.5 further breaks down the above results by showing the *Reject-Some* rate for different countries. To provide a solid picture, we again limit the analysis to the countries for which we record at least 10 websites with at least 10 interactions in both periods – showing the first 15 countries by descending *Reject-Some* rate in *Period B*. The figure confirms the previous results. In *Period A*, we do not observe significant differences in the *Reject-Some* rate between GDPR (France, Romania, Germany, Spain and the UK) and non-GDPR countries. Indeed, confidence intervals overlap. In *Period B*, conversely, with the insertion of the additional `Reject All` button, we observe a significant increase in all GDPR-regulated countries, from a $\sim 3.5\times$ in the UK and Germany ($\sim 6\%$ to $\sim 23\%$) to more than $7\times$ in Portugal (2.14% to 15.34%). As expected, there are no significant variations in other, non GDPR-regulated countries.

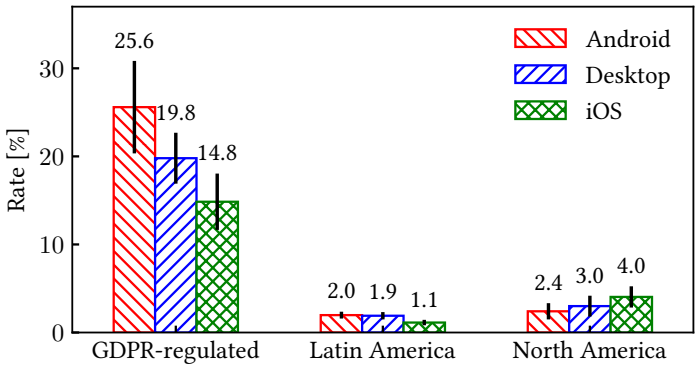


Fig. 3.6 *Reject-Some* rates according to users' device.

Table 3.3 Relative distribution of used devices per region of connection.

Region	Android	iOS	Desktop
GDPR-regulated	41.6%	40.5%	17.9%
Latin America	40.6%	38.9%	20.5%
North America	30.0%	44.8%	25.2%

3.2.3 User device type

We now move on to analyze the differences between users browsing the Web with different types of devices. To this end, we categorize each interaction based on the client-side `User-Agent` HTTP header, to obtain the operating system (OS) of the user's device. Considering that the experience of navigating websites is not greatly affected by OS when using a PC, we group Windows, Mac OS, Linux and other operating systems under the same *Desktop* category. Conversely, we divide the mobile landscape into two main major categories: *Android* and *iOS*. Overall, *Desktop*, *iOS*, *Android* represent the 21%, 39% and 40% of the entries, respectively. Other mobile OSes are present in the dataset, but their volume is so low that we neglect them. The region-wise device shares are overall homogeneous across the regions and are reported in Table 3.3.

In Figure 3.6 we show the *Reject-Some* rate separately by OS. We target *Period B* because our dataset contains the `User-Agent` field only after August 25, 2022. For the non-GDPR regions, we choose North and Latin America as they are the origin of the largest amount of interactions (see Table 3.2). We include a website in the macro-average only if it collected at least 10 interaction in the target (website,

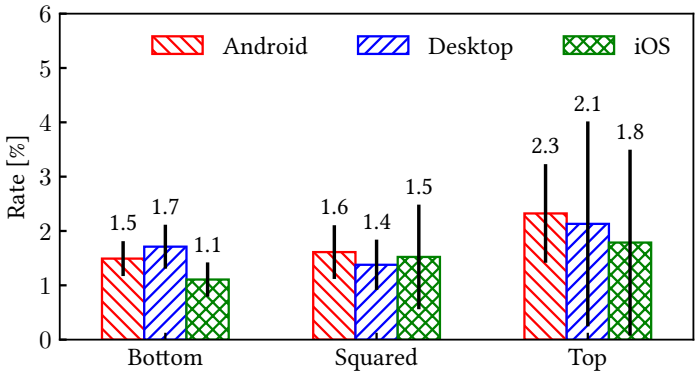


Fig. 3.7 *Reject-Some* rates according to the position of the screen where the banner appears.

region) couple. Overall, the figure confirms the large difference between regions that we discussed in Chapter 3.2.2. Moreover, it surprisingly shows that, in countries subject to the GDPR, Android users are more likely to reject some cookies than iOS users – see the first box group in Figure 3.6. One could argue that iOS users feel safer than Android users due to Apple's efforts to enforce and communicate privacy-preserving technologies on its devices, but the data at our disposal do not allow us to prove any hypothesis about it. This is nonetheless an interesting finding, which offers a stimulating question for future work. The same consideration holds for Latin America, while in North America, the averages have reversed roles. We limit ourselves to observing these figures, while the search for the causes of this behavior requires other data and possibly controlled experiments.

3.2.4 Banner size and position

We then investigate whether the shape and dimension of the banner presented to the users impacts their behavior. Figure 3.7 shows the *Reject-Some* rate according to the type of banner presented to the user. Let us recall that three types of banners are offered by the CMP: a top-screen long and narrow banner (identified as Top), a bottom-screen long and narrow banner (Bottom), and a bottom-left square banner (Squared). Here we limit the analysis to *Period B* and areas that are not regulated by the GDPR and for which we have a larger number of interactions. In fact, the websites that use the Squared banner account for about 5%. Thus, they received a number of interactions by GDPR countries that does not allow to draw solid conclusions.

Table 3.4 Breakdown of partial accept among categories of cookies.

Category of Cookies	Acceptance Rate
Necessary	$100.00 \pm 0.00\%$
Statistics	$58.42 \pm 3.19\%$
Preferences	$47.14 \pm 3.23\%$
Marketing	$22.10 \pm 2.68\%$

Overall, Figure 3.7 seems to suggest there are not significant differences in the way users interact with the banner with respect to its shape and position. Few websites implement the Top banner, resulting in a large confidence interval.

Unfortunately, our data do not allow to track the behaviour and the volume of users that *neglect* the Privacy Banner – i.e., do not interact at all with it. Thus, we cannot measure whether the fraction of users interacting over the total visitors differs according to the different position or shape of the banner.

3.2.5 Other behaviours

As observed in Chapter 3.2.2, users accessing the `Custom Permissions` screen are in the order of a few percentage points. This confirms that the majority of users do not bother taking precautions for their privacy if this requires more than one click. In this section, we characterize the behavior of users dealing with advanced options.

Cherry-picking cookies

By clicking on the `Custom Permissions` button, users are offered the possibility to give separate consent for different types of cookies. Out of 4 M user interactions of our dataset, only 647 times users customized consent for different cookie categories – i.e., provided a *Custom* consent. For completeness, in Table 3.4, we show the acceptance rate for each category, uniquely for the 647 entries that correspond to *Custom* consent. To evaluate confidence interval, we consider that the proportion of users that accept a category of Cookies (e.g., Statistics) is an unbiased estimator of the probability p of a Bernoulli random variable. Assuming that all the interactions are independent repetitions of such random variable, we obtain the number of successes

of a binomial random variable. We thus use binomial proportion confidence intervals, with a confidence level of 90%.

Necessary cookies, represented in the first row of the table, are mandatory and, therefore, cannot be disabled by the user as they are required for website operation. Statistics cookies are the most accepted (58% of cases). These cookies are related to analytics services that account for the number of accesses to the website and monitor performance. Preference cookies, used to recognize users when they return to the website, are accepted to a similar extent (47%). Finally, Marketing cookies are most often rejected. Only 22% of users accepted them. These cookies include web trackers and advertising platforms. Users tend to avoid them, and we can guess that they are perceived as the most privacy intrusive.

Visualizing policies

We finally quantify the number of users who access the text of the policies regulating the use of personal data in a website. Indeed, websites must offer the possibility to access this information, and the CMP includes links to Cookie and Privacy Policies. Unless the website implements some customization, the Cookie Policy includes a brief explanation on the concept of cookie, information on the categories of cookies collected by the CMP (Necessary, Statistics, Preferences, Marketing) and their purpose. The Cookie Policy is presented as a small pop up (305 word in its default formulation, in English) and the users do not leave the page they are visiting. Conversely, clicking on the Privacy Policy opens a new webpage which can be either hosted on the website or served by the CMP. The Privacy Policy contains information about the use of personal data at large, of which the cookies represent only a subsection. The policy includes, among the rest, information about the purpose of data collection, the parties with which said data might be shared, the retention policy of the data, etc.

Our dataset records all clicks on the Cookie Policy. Those on the Privacy Policy are tracked only if the policy is hosted by the CMP, so we restrict our analysis to approximately one-fourth of the total interactions. Again, due to the low number of interactions of this type, we do not show the website-wise macro-average but provide the overall numbers directly in Table 3.5, while the confidence interval are again calculated using binomial proportion. The number of interactions in which a user either clicks on at least one of the links is very low: $2,469$, 0.24% of the total. Users

Table 3.5 Number interactions related to users clicking or not on the Cookie Policy (CP) and the Privacy Policy (PP). The last column indicates, the *Reject-Some* rate for the given set of interactions.

PP clicks	**CP clicks**	**Interactions**	***Reject-Some* rate**
Yes	Yes	349	$7.45 \pm 2.75\%$
Yes	No	944	$6.04 \pm 1.52\%$
No	Yes	1 176	$3.57 \pm 1.06\%$
No	No	1 011 737	$1.01 \pm 0.02\%$

who decide to read (or at least visualize) the policies appear more careful about their privacy: those who click on both policies record a 7.45% *Reject-Some* rate, while users who do not visualize any account for a value of only 1.01%. Although we cannot prove that the *Reject-Some* rate increases *because* users read the policies, there is at least a sizeable correlation between users' interest in the policies and their unconditional *Accepted-All* rate.

3.3 *Priv-Accept* design and testing

As we have extensively shown in Chapter 3.2, the percentage of users accepting the privacy policies offered by the websites is a large fraction of the total. This fact challenges the commonly accepted approach to automatically crawl websites to measure the Web ecosystem on privacy [21, 52, 19, 53, 20, 54, 49, 22, 23, 55, 20, 57, 58] and performance [60, 61, 63, 62, 64, 65, 68, 66, 69]. These measurements are typically carried out with headless browsers that access webpages and automatize the collection of metadata and statistics. However, today, these measurements could be biased and unrealistic, with the crawler observing possibly very different content than what a user would get after accepting the privacy policies. In fact, the Privacy Banners may hide the actual page content, and the browser may load additional content only after the privacy policy acceptance.

We explicitly engineer *Priv-Accept* to fully automate the visit to websites and collect statistics. The key element of *Priv-Accept* is its ability to identify the presence of a Privacy Banner and automatically accept privacy policies. We aim at a practical and effective approach to accept privacy policies through the offered button.

To illustrate *Priv-Accept* operation, consider again Figure 2.1. A large Privacy Banner appears on the first visit, and the user shall click on the "Got it" button to access the webpage content. *Priv-Accept* has to locate this button and click on it automatically. As a result, the website starts loading advertisements and contacting trackers in the background. We refer to these two types of visits as *Before-Accept* and *After-Accept* in the remainder of the chapter.

We implement *Priv-Accept* using the Selenium browser automation tool [59], the de-facto standard for browser automation, using Google Chrome as browser. Given a target URL, *Priv-Accept* carries out the following tasks:

1. It navigates to the URL with a fresh browser profile, i.e., with an empty cache and cookie storage. This makes the visit the equivalent of a *Before-Accept* to the website.

2. It inspects the Document Object Model (DOM) of the rendered webpage to find a possible *Accept-button* in a Privacy Banner. For this, it matches a list of keywords on the text of each node of the DOM. We identify an *Accept-button* if we exactly match any of these keywords. For robustness, we remove leading/trailing/repeated blank characters and the match is performed ignoring the case. We do not use stemming, lemmatization or other techniques for text processing given the specificity of the words to match and the need to support multiple languages.

3. If *Priv-Accept* finds the *Accept-button*, it clicks on the corresponding DOM element (typically a `<button>`, `<href>` or `<span>` element) to accept the privacy policy and logs the success acceptance.

In the beginning, we built *Priv-Accept* to look for accept buttons through CSS selectors combined with keywords as done in [70] and popular add-ons. However, we soon observed that this methodology was too fragile as the use of selectors is strongly CMP-specific and highly customizable by webmasters. The keyword-based approach eases the generalization of the solution. Considering complexity, *Priv-Accept* adds marginal overhead to the time required to visit a webpage. Only for very complex webpages, iterating through all DOM elements may require some time, but this is still less than the time needed to load and render the webpage by the browser.

During each visit, *Priv-Accept* stores metadata regarding the whole process in a JSON log file. It includes details on all HTTP transactions and installed cookies.

Moreover, it optionally takes screenshots of the webpage during the various phases to allow manual verification.

Priv-Accept is highly customizable and offers the user various features. It lets the user customize the declared `User-Agent` and browser language (in the `Accept-Language` headers). Important to our analysis, it can be configured to run a:

- *Warm-up visit*: to populate the browser cache.
- *Before-Accept*: to collect statistics on the webpage before accepting the privacy policy, as a Naive Crawler would do.
- *After-Accept*: to collect statistics on the webpage as it appears after accepting the privacy policy (if an *Accept-button* is found).
- *Additional-Visits*: to a number of webpages of the same website, randomly choosing among the internal links.[5] We visit internal pages both if *Priv-Accept* finds the *Accept-button* and if it does not.

For each page visit, *Priv-Accept* collect several metadata. Considering QoE metrics, here we focus on the Page Load Time, or *OnLoad* time [108]. It allows us to compare the webpage rendering performance with and without privacy policy acceptance. It is simpler and faster to compute than the SpeedIndex [109], allowing large scale measurements. Notice that we neglect metrics that are not affected by the presence of a Privacy Banner, such as the Time-to-first-byte (TTFB).

Notice that the *After-Accept* visit can only occur with a warm browser cache in real cases since the browser would have first to complete the *Before-Accept* visit. To fairly compare a *Before-Accept* and *After-Accept*, in our experiments we run a preliminary *Warm-up visit* before the *Before-Accept* to fill the browser cache. This lets us appreciate the eventual extra time to load additional components and fairly compare the *OnLoad* on the two visits with the hot cache. Alternatively, *Priv-Accept* can erase the HTTP cache and clean the socket pool upon each visit to compare webpage performance with a cold cache.

Priv-Accept follows possible redirects during the visits and cases when a script triggers a reload of the webpage. This allows us to manage cases in which the Privacy

[5]We define internal links as those having the same Fully Qualified Domain Name as the visited website.

Banner is hosted on a separate specific landing page than the actual website home page. At last, to limit the impact of random delay due to webpage download and rendering, *Priv-Accept* uses quite conservative timeouts before eventually abort the visit. In detail, the DOM inspection starts 5 seconds after the *OnLoad* event. While this clearly slows down the visit of multiple webpages, it maximizes the success rate.

To allow large-scale measurement campaigns, we containerize *Priv-Accept* using the Docker container engine [110]. In the containerized version, we use Google Chrome version 89 in headless mode and force it to use a standard `User-Agent` instead of the pre-defined `ChromeHeadless`.[6]

3.3.1 Keyword Selection and Validation

At the core of *Priv-Accept* there is the list of keywords to be matched against the webpage content to localize the clickable DOM element for accepting the privacy policy. We thoroughly build this list manually in an iterative way. To handle different languages, we build a list that includes keywords for each country we are interested in. For this work, we focus on 5 European countries, namely France, Germany, Italy, Spain, UK[7], plus the US – which we use as an example of a large, extra-EU country were privacy laws are in force. For each country, we pick the most popular websites according to the Similarweb lists [111], a website-ranking service analogous to Alexa.

First Round - keyword extraction from top websites

In the first round, for each of the 5 countries, we consider the top-200 websites that have a Privacy Banner. We randomly choose half of these websites and manually visit them (from Europe) to extract the accept keyword. In total, we visit 500 websites and identify 186 unique keywords. We next instruct *Priv-Accept* to visit the other half of websites and let it accept the privacy policy, if found. For those where it fails (233 cases), we manually visit them to check i) if they have a Privacy Banner, and ii) eventually to extract new keywords. With this, we identify 36 new keywords, 222 in total. During these steps, we also check that the tool correctly accepts the policy.

[6]The containerized version is available on Docker Hub as *martino90/priv-accept*.

[7]In January 2021 UK has enforced the UK GDPR, with practically identical requirements.

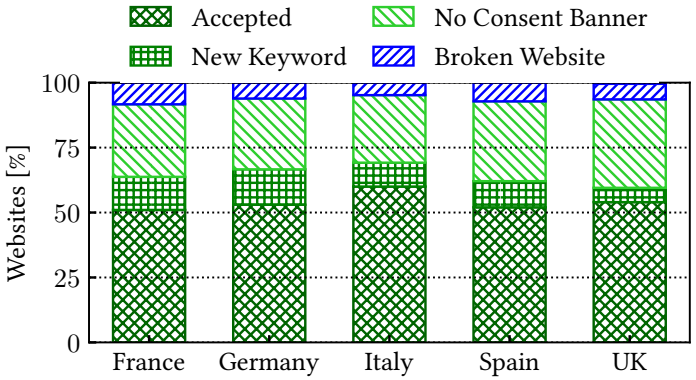


Fig. 3.8 Validation results of *Priv-Accept* over 200 randomly picked websites per country.

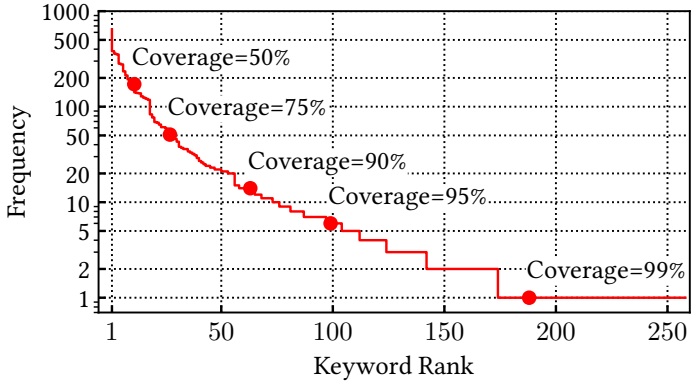


Fig. 3.9 Frequency of the *Priv-Accept* keywords, with indication of the coverage at different points.

Second Round - testing and keyword increase

To evaluate the accuracy of *Priv-Accept* in the wild, we next consider 200 new random websites for each country from the Similarweb lists, 1,000 websites in total. We let *Priv-Accept* visit them and manually check the subset of 448 websites for which *Priv-Accept* did not find (and accepted) a privacy policy. We depict the results in Figure 3.8. *Priv-Accept* can accept the privacy policy in more than half of websites, independently from the language. In $6-14\%$ of cases, we find 36 new keywords – that we promptly add to our list. Interestingly, we find a non-negligible portion of websites ($26-30\%$) that do not present any Privacy Banner. At last, *Priv-Accept* fails to accept privacy in only $5-8\%$ of cases. Investigating further, this is due

to some non-standard behavior of the webpage when accessed in headless mode. For instance, some websites present a CAPTCHA when they detect an automated visit; other websites return a blank webpage. This is a common problem for any crawler-based measurement study [112]. For completeness, cases of *False Positives* – i.e., *Priv-Accept* clicking on a wrong DOM element – are possible, although we have not observed any in our manual validation tests.

At the end of the keyword list building phases, we collect a total of $258(186+36+36)$ keywords obtained by manually visiting $1181(500+233+448)$ websites, covering 6 languages.[8] In Figure 3.9, we show the distribution of keyword appearance frequency across the entire set of 12,277 Similarweb websites (see Chapter 3.3.3 for details on this list). The most common keyword is the string "Ok". Red dots indicate the portion of websites covered by the top-N keywords – i.e., the coverage of the top-N words. The top keywords are very common (note the logarithmic scale on the y-axis), with the top-10 that cover half of the websites. The top-98 keywords cover 95% of the websites, while the remaining appear less than 10 times each in the whole website set. Clearly, we expect the list of keywords to naturally grow as the tail of the Figure 3.9 suggests. Notice indeed that more than 80 keywords have been found on a single website. Curiously, we find complex strings like "I'm fine with this" or "Alle auswählen, weiterlesen und unsere arbeit unterstützen"[9].

3.3.2 *Priv-Accept* vs. Consent-O-Matic

We compare the effectiveness of *Priv-Accept* with Consent-O-Matic [76], the most mature browser plugin designed to offer/deny consent to privacy policies automatically. Unlike our tool, Consent-O-Matic exploits the presence of popular Consent Management Platforms (CMP), services that take care of the management of users' choices on behalf of the website. At the time of writing, Consent-O-Matic allows managing Privacy Banners for 35 CMPs. To gauge its performance, we visit the top-100 most popular websites with a Privacy Banner for the 5 countries using a Chrome browser with the Consent-O-Matic plugin enabled. Consent-O-Matic accepts the privacy policies in less than 35% of websites with Privacy Banner, and as little as 17% and 20% for websites in Italy and UK, respectively. Here *Priv-Accept* accepts the privacy policies on all websites by construction.

[8] In Spain, some websites are in Catalan, rather than in Spanish.

[9] Which translates to "Select all, keep reading and support our work".

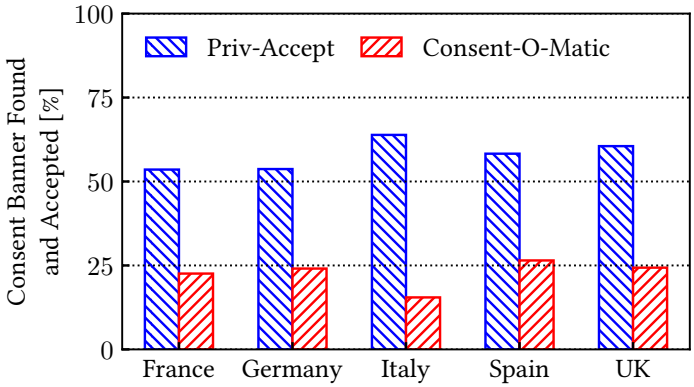


Fig. 3.10 Privacy policy acceptance rate of *Priv-Accept* and Consent-O-Matic on 100 websites per country.

We then run a second experiment considering another set of 100 websites randomly picked from the Similarweb per country lists. We visit each website with *Priv-Accept* and a Consent-O-Matic-enabled browser. Figure 3.10 summarizes the comparison. *Priv-Accept* accepts the privacy policies in more than 50% of websites, more than twice the success rate of Consent-O-Matic. These results are in line with those of Figure 3.8. The remaining websites may not have a Privacy Banner, fail to load, or use an unknown keyword. This testifies that the customization of Privacy Banners makes it difficult to engineer a generic and simple solution. The keyword-based strategy results more robust than the CMP-based approach (with similar complexity in curating the lists).

3.3.3 Dataset and Tracker list

In the following, we use *Priv-Accept* to check the impact of using *Priv-Accept* when doing large web measurement experiments. We targets a large set of websites popular in France, Germany, Italy, Spain and US, using a test server located in our university campus. For each of the 6 countries, we use the Similarweb lists to select the top-100 websites from 24 different categories – see Figure 3.16. These are the top-level unique categories listed in the Similarweb page [113]. In total, we include 12,277 unique websites to visit (as the lists in different countries partially overlap). When visiting websites of a given country, we set the `Accept-Language` header to indicate

the appropriate locale and country language. This behavior can be configured in the *Priv-Accept* configuration to allow further experimentation.

We run *Priv-Accept* on a single high-end server running 16 parallel instances to speed up the crawl. We instrument it to run a *test sequence*, which consists in a *Warm-up visit*, *Before-Accept* and *After-Accept* to the landing page, followed by *Additional-Visits* to 5 randomly chosen internal pages – previous studies indeed show that internal and landing pages have different properties [56]. For each website, we repeat the test sequence 5 times, randomizing the order of websites to visit in each repetition. Our main experimental campaign took place for two weeks on April 2021.

We run additional measurement campaigns to investigate specific aspects. To understand whether Privacy Banners appear or have a different impact depending on the visitor location, we repeat the above experiments using servers located in the US, Brazil and Japan. We use Amazon Web Services to deploy on-demand servers on the desired availability zone. Here, we aim to check if websites behave differently based on the location of the visitors. Since we are using cloud servers, targeted websites may wrongly recognise the test machines as not regular users and located them in a generic or wrong country. While we cannot check this, we verified that the two most popular commercial IP location databases (IP2Location[10] and MaxMind[11]) map the IP addresses of our crawlers to the correct country.

To offer a view on a larger number of websites, we visit the top-100,000 websites according to the Tranco list [114]. Unfortunately, the Tranco list does not offer a per-category and per-country rank. We run two separate test sequences: with warm caches, doing (i) *Warm-up visit*, (ii) *Before-Accept*, and (iii) *After-Accept*. And with cold caches, (i) *Before-Accept*, (ii) erase HTTP cache and clean socket pool and (iii) *After-Accept*. Following this procedure, we ensure a fair comparison between *Before-Accept* and *After-Accept* in the two scenarios. Recall that *Priv-Accept* allows one to generate any combination of test sequence with warm/cold cache.

To observe how the presence of trackers changes, we rely on publicly-available lists provided by Whotracksme [115] (a tracking-related open-data provider), EasyPrivacy [116] (one of the lists at the core of AdBlock tracker-blocking strategy) and AdGuard [117] (a popular ad-blocking tool). For robustness, we merge the three

[10] https://www.ip2location.com/
[11] https://www.maxmind.com/

lists and consider as a potential tracker any third-party domain that appear in at least two lists. In total, we obtain 1,497 domains that we consider tracking services.[12] We finally record the presence of a tracker during a visit if the webpage embeds an object from a tracking domain, and the latter installs a cookie with a lifetime longer than one month [24] – commonly referred to as *profiling cookie*. As such, we divide the HTTP transactions carried out during a visit in:

- First-Party: objects from the same domain of the target webpage.
- Third-Party: objects from a different domain than the target webpage.
- Trackers: objects from a Third-Party that is a tracking domain and sets a profiling cookie.

3.4 Impact on Tracking

In this section of the chapter, we characterize how the Web tracking ecosystem changes if observed with or without accepting the privacy policies. We break down results by Third-Party/Tracker, by country and website category.

3.4.1 Third-Party and Tracker Pervasiveness

We first study the pervasiveness of Third-Parties and Trackers and check how it varies when we measure it in a *Before-Accept* or *After-Accept*. *Priv-Accept* found and accepted a Consent Banner on 63.2% of websites. Here, we aim at quantifying the impact of privacy policy acceptance on European websites (10,542 in total) and we exclude those websites exclusively popular in the US.

We first detail the top-15 most pervasive Third-Parties in Figure 3.11. The GDPR mandates to obtain informed consent before starting to collect any personal data. As such, Third-Parties may be seen as possibly offending services if activated before accepting the privacy policy.[13] With little surprise, the most pervasive Third-Party

[12] In the following, we identify them with their *second-level domain name* – i.e., a hostname truncated after the second label. We handle the case of two-label country code TLDs such as `co.uk`.

[13] Here, we do not enter into the debate of what can be considered a Tracker.

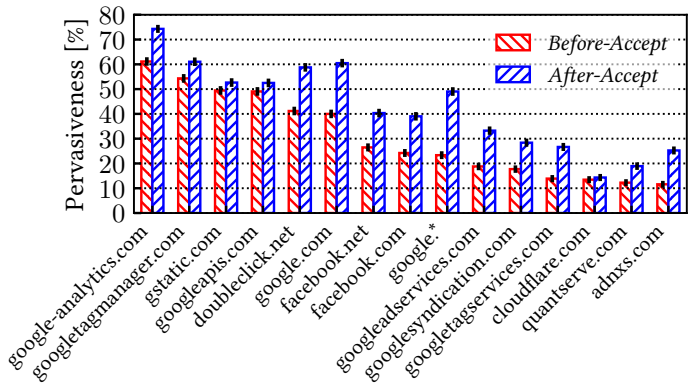


Fig. 3.11 Pervasiveness of the top-15 Third-Parties (percentage of sites they are in) on 10 542 websites popular in Europe.

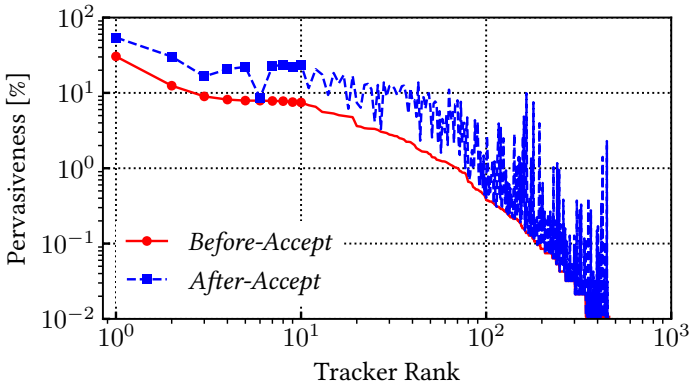


Fig. 3.12 Pervasiveness of the 342 identified Trackers (percentage of sites they are in) in 10 542 websites popular in Europe.

is `google-analytics.com`. It grows from 61% to 74% in popularity on the *After-Accept*. This value is surprisingly similar to what Metwalley *et al.* [118] found in 2016, when they found `google-analytics.com` appearing in 71% of websites. The growth is also sizeable for other Google services such as `googleadservices.com` and `googlesyndication.com`. Conversely, domains belonging to Content Delivery Networks, such as `cloudflare.com` and `cloudflare.net` do not increase their pervasiveness on the *After-Accept*, likely being not included in the mechanisms of Consent Banners. Interestingly, only 3 out of the top-15 Third-Parties are Trackers – i.e., present in our tracker list and setting a persistent cookie. `doubleclick.net` and `facebook.com` are the most popular ones, with pervasiveness growing from 41% to 58% and from 24% to 39% on the *After-Accept*, respectively. They are present in more than twice the number of websites than their first competitor (`quantserve.com`). In Figure 3.11, we also report 95% confidence intervals. It results that the sample proportion (in percentage) of pervasiveness of Third-Parties is an unbiased estimator of the probability p of a Bernoulli random variable. Therefore, by repeating a number of occurrences of a Bernoulli random variable equal to the number of samples, we obtain the number of successes of a binomial random variable. The confidence intervals become the classical binomial proportion confidence intervals. For the sake of completeness, we report error bars also in the following plots. Note, that, given the large number of samples, the confidence intervals are very narrow and not overlapping between *Before-Accept* and *After-Accept*, except for the case of `cloudflare.com`.

Focusing now on Trackers only, we show their pervasiveness in Figure 3.12. We count 342 of them. The red curve shows the pervasiveness on the *Before-Accept*, which is what a naive crawler would report. The blue curve shows how the figure changes on the *After-Accept*. The Trackers on the x-axis are sorted in descending order according to their pervasiveness on the *Before-Accept*– hence the *Before-Accept* curve is monotonically decreasing, while the *After-Accept* is not. Note that the figure has log-log axes to better show the large variability of Tracker popularity. The increase in pervasiveness is general and includes both popular and infrequent Trackers, reaching one order of magnitude in a some cases. On the *After-Accept*, the number of Trackers that are present on 1% or more of websites grows from 40 to 90.

Here, the Spearman's rank correlation is 0.90, indicating that the Tracker popularity order is approximately the same before and after the privacy policy acceptance. The difference is that their pervasiveness increases.

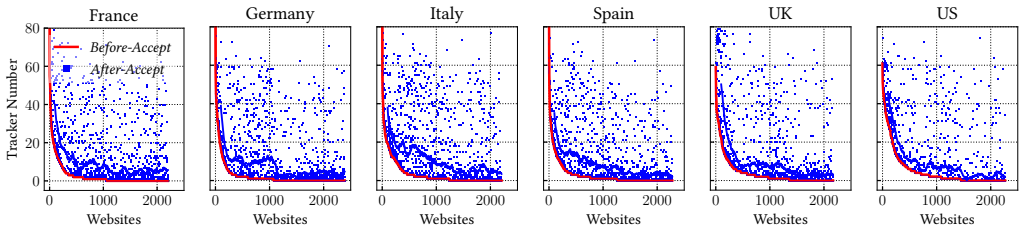


Fig. 3.13 Trackers per website seen on the landing page.

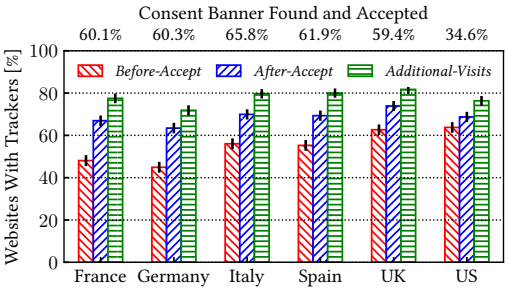
As it emerges from Figure 3.12, many Trackers are widespread even on the *Before-Accept*. This hints at a possibly wrong implementation of the GPDR regulation, which mandates acquiring the visitor's explicit consent before activating any tracking mechanisms. To be precise, the presence of Trackers on the *Before-Accept* does not necessarily entail a violation of the law. An analysis of the most popular cookies reveals the presence of test cookies during the *Before-Accept* using a form similar to `test_cookie = CheckForPermission`. Google Analytics is a notable example. These cookies are just a check for the possibility of installing profiling cookies upon the user's acceptance. It is thus possible that the *Before-Accept* pervasiveness of some Trackers includes cases in which only test cookies are actually used (curiously with expiration date longer than a month). Here we limit to observe that often Trackers set some (potentially) profiling cookies even on the *Before-Accept*.

3.4.2 Breakdown on Websites

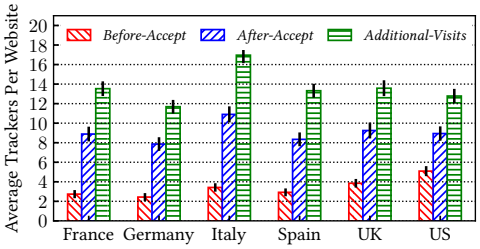
We now detail the impact of accepting privacy policies on the number of Trackers found in each website, breaking down our results by country and website category.

Analysis by country

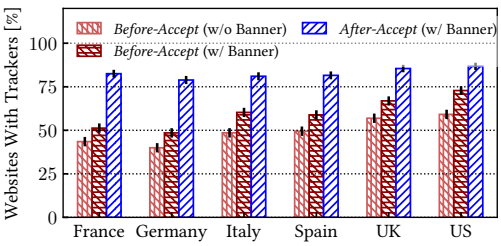
Figure 3.13 shows websites (top 2,500 per country) sorted in descending order by the number of contacted Trackers as measured in the *Before-Accept* (red curve). This number tends to grow on the *After-Accept* (blue points), where we observe some websites that present 50-70 more Trackers. To increase readability, in Figure 3.13, the blue line reports the moving average (with a 100 window) of the number of contacted Trackers on the *After-Accept*. Curiously, some websites that already include Trackers in the *Before-Accept* include more Trackers in the *After-Accept*. This again may hint



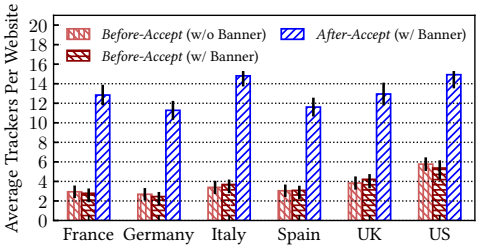
(a) Percentage of websites embedding Trackers. The top *x*-axis details the fraction of websites in such country where *Priv-Accept* found and accepted privacy policies.



(b) Average number of Trackers per website.



(c) Percentage of websites embedding Trackers, splitting websites with and without a Consent Banner.



(d) Average number of Trackers per website, splitting websites with and without a Consent Banner.

Fig. 3.14 Tracker penetration during different phases of a browsing sessions (top 2,500 websites per country).

at a wrong implementation of the Consent Banner, which fails to hinder the presence of offending Trackers. The increase is less remarkable for US-popular websites – mainly due to the less widespread presence of Consent Banners.

To better quantify Tracker presence, we show the fraction of websites containing at least one Tracker in Figure 3.14a. As in Figure 3.11, we report 95% confidence interval on these sample proportions. About 50% of websites popular in European countries already include at least one Tracker on *Before-Accept*. This happens more frequently in the UK (63%) and less often in Germany (44%). Again, note that a website embedding a Tracker on the *Before-Accept* does not necessarily represent a violation of the GDPR, even if this can often be the case [24]. Interestingly, in the US this figure is higher than in European countries. Recalling that the probability of encountering a Consent Banner in the US is lower, this hints at a positive effect of the GDPR on popular European websites. The percentage of websites containing Trackers in the *After-Accept* grows for all European countries from a $+11\%$ increase in the UK to $+20\%$ for Germany. Confidence intervals never overlap. This increase is moderate ($+5\%$) in the US, given the lower fraction of those websites having a Consent Banner. We complete this analysis by reporting how this fraction increases when performing 5 *Additional-Visits* as recommended in [56]. Our results confirm this need, with the chance to observe at least one Tracker that further grows by 5%-10% in *Additional-Visits* when compared to the *After-Accept*. Note that, considering each country, none of the confidence intervals overlap between *Before-Accept* and *After-Accept* and between *After-Accept* and *Additional-Visits*.

We next investigate the quantity of Trackers contacted while visiting websites in Figure 3.14b, which shows the average number of Trackers contacted on the websites, separately by country. Also in this case we report 95% confidence intervals. The sample mean is an unbiased estimator of the true mean, and we can derive confidence intervals through central limit theorem. For all countries, the average amount of Trackers more than doubles on the *After-Accept*, and performing *Additional-Visits* further increases this figure (with non-overlapping confidence intervals). In Italy, for instance, this figure grows by a factor of 4 when comparing *Before-Accept* and *Additional-Visits*. As previously noted, the behavior of US-popular websites differs from the European: before acceptance, the number of Trackers is already higher than in popular European websites, while it is comparable after. This hints that popular websites in the United States may be less receptive to GDPR indications. On the opposite side, German-popular websites appear to be the most observant of the

regulations, installing Trackers only upon accepting the privacy policies. Afterwards, they reach levels comparable to the other countries. In summary, European websites use the same quantity of Trackers as US ones, although they are often contacted only after accepting the privacy policy.

To appreciate the variation in the number of Trackers for those websites implementing a Consent Banner, we deepen the analysis by showing separately websites for which *Priv-Accept* has found (or not) a Consent Banner. Our goal is to show how Tracker number varies on the *Before-Accept* and *After-Accept* for those websites implementing the Consent Banner. Figure 3.14c shows the percentage of websites with at least one Tracker, and Figure 3.14d shows the number of Trackers per website. The dark red bars and blue bars show results on the *Before-Accept* and *After-Accept* for those websites where *Priv-Accept found* a Consent Banner. As before, the increase of Trackers is sizeable. For completeness, the light red bars report the same measure for those websites where *Priv-Accept did not* find any Consent Banner.

We finally observe that the probabilistic nature of Web tracking and bidding mechanisms results in a different number of Trackers contacted at each visit. To obtain the most reliable measurements, we test each website 5 times, each time visiting 5 internal pages. We note that measuring the fraction of websites containing at least one Tracker (as in Figure 3.14a) is moderately impacted by the number of tests. Indeed, when considering a single *After-Accept* per website, overall, we find 69.1% of them containing one (or more) Trackers. Repeating 5 times the test and considering whether we find at least one Tracker among all visits, this percentage increases only to 70.0%. Similarly, the average number of Trackers (as in Figure 3.14b), increases from 6.5 to 7.8. In Figure 3.15, we show how two macroscopic tracking measurements vary with different number of repeated visits for each website. The blue line in the figure shows the fraction of websites that contain at least one Tracker when measured with an increasing number of test repetitions.

Analysis by category

We now break down the picture by category, showing the results in Figure 3.16. We explicitly target websites of 24 categories, each containing the top-100 websites for the considered countries. We sort categories from the highest to the lowest percentage of websites with Trackers in *Before-Accept*. 95% confidence intervals are reported on each bar.

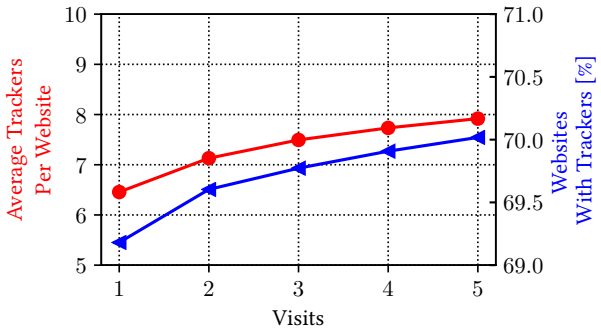
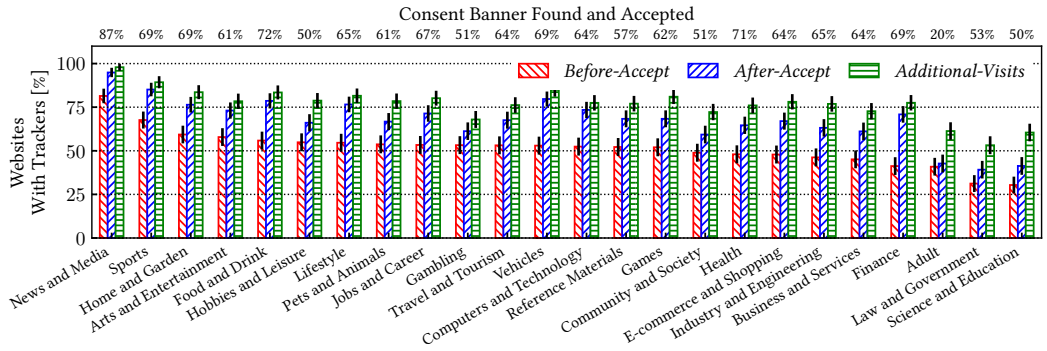


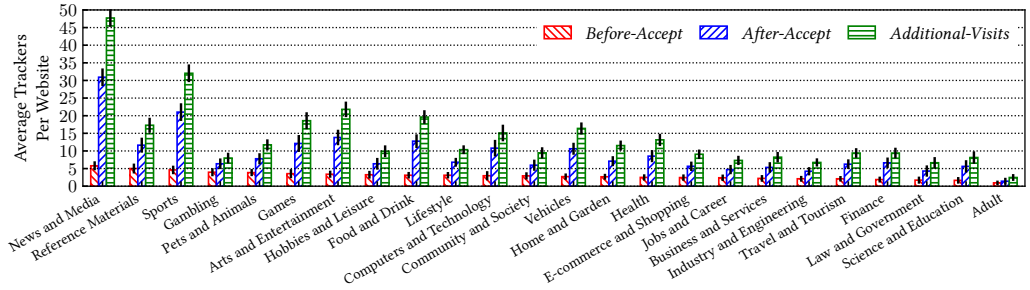
Fig. 3.15 Variation of tracker number with different numbers of repeated visits.

Starting from Figure 3.16a, we report the percentage of websites of a given category that contain at least one Tracker. As before, there is a large increase from *Before-Accept* to *After-Accept*. Exceptions are the *Adult*, *Law and Government* and *Gambling* categories, where the confidence intervals overlap. For *Adult* this is likely due to the low number of websites with Consent Banners (20%) and confirms the peculiarity of the tracking ecosystem on Adult websites [70]. As previously observed in Figure 3.14a, performing *Additional-Visits* further increases the chance of encountering at least one Tracker, even though in this case the increase is limited and we observe some overlaps between *After-Accept* and *Additional-Visits* confidence intervals.

Moving to the number of trackers per website shown in Figure 3.16b, we observe large increase in the *After-Accept* case, confirming that most Trackers appear only after the user accepts the privacy policies and when visiting internal pages. Here, differences across categories are all pronounced, with those categories that heavily depend on advertisements (*News and Media*, *Sports*, *Games*, *Arts and Entertainment*) that have to rely on a large number of Trackers to support behavioral advertisements. This is noticeable already on the *Before-Accept*. For example, access to a *News* website leads to contact 5.7 Trackers on average in *Before-Accept*. Here, *Priv-Accept* successfully accepts the privacy policies in 87% of cases. Indeed, being *News* websites very popular, they tend to correctly implement the privacy regulations and to show a well-configured Consent Banner. Upon acceptance, suddenly, the number of Trackers becomes almost 6 times higher (30.9 for *News*) and 9 times higher when doing *Additional-Visits* (47.7 trackers on average). For *Sport*, *Food and Drink*



(a) Percentage of websites embedding Trackers. The top *x*-axis details the fraction of websites in such category where *Priv-Accept* found and accepted privacy policies.



(b) Average number of Trackers per website.

Fig. 3.16 Trackers penetration and number on websites (top 2,500 per country) during different phases of a browsing session, separately by category.

and *Arts and Entertainment* the average number of Trackers more than triples in *After-Accept*. Only for the *Adult* category confidence intervals overlap.

These numbers are particularly interesting if read in the perspective of recent works. Englehardt *et al.* [53], in 2016, measured an average of 35 Trackers per website on News websites. In 2021, we find similar numbers (30.9) on the *After-Accept*, while, due to the spread of Consent Banners, on the *Before-Accept* we would only find 5.7, on average. On Sport category, Englehardt *et al.* [53] measured 27 Trackers per website. In 2021, we find 21.0 on the *After-Accept*, while only 4.6 on the *Before-Accept*. These results well highlight the need for correctly handling the Consent Banners to observe the extensiveness of web tracking. In a nutshell, thanks to *Priv-Accept*, we obtain the fundamentally different figure in the *After-Accept* and *Additional-Visits*.

The case of *Adult* websites is worth a specific comment. *Priv-Accept* finds the Consent Banner on only 20% of them, and a manual check on 50 of them confirms that the large majority of them do not offer any Consent Banner. Tracking is also limited upon acceptance, and the confidence intervals between *Before-Accept* and *After-Accept* even overlap. Similar results were previously found by Vallina *et al.* [70], where the authors suggest that the specialized pornographic advertisement ecosystem may cause this behavior: usually, trackers and advertisers related to pornographic websites do not operate outside of them – often evading popular tracker lists.

3.4.3 Visits from Outside Europe

We now consider additional measurement campaigns using crawling servers in the Amazon AWS data centers located in the US (Ohio and California), Japan and Brazil. Figure 3.17 summarizes our findings. First, notice how *Priv-Accept* accepted privacy policies on around 10% fewer websites (about $1\,150 - 1\,200$) when run from outside Europe, as reported on top x-labels. Checking the screenshot taken by *Priv-Accept* during the visit on a random subset of these websites, we confirm that no Consent Banner is displayed. We can conclude that some websites customize the Consent Banners based on visitors' properties, such as their location. If the visit comes from not EU country, no Consent Banner is shown.

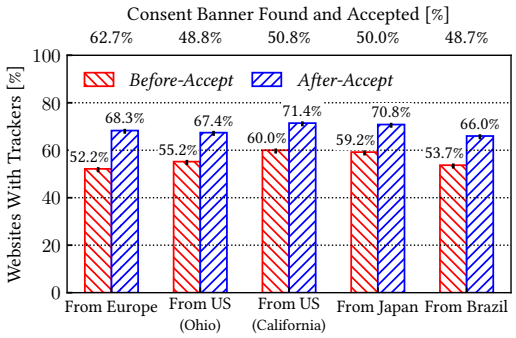
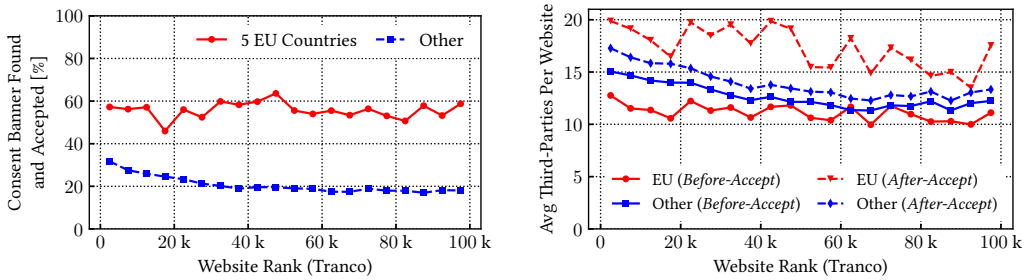


Fig. 3.17 Websites with Trackers (12 277 from the Similarweb lists) when crawling from different countries.



(a) Percentage of websites with a Consent Banner. (b) Average number of Third-Parties per website.

Fig. 3.18 Percentage of websites with a Consent Banner and average Third-Parties per website over the top-100 k websites in Tranco list, computed every 5,000 websites in the rank.

This different behaviour of websites affects also the statistics of the fraction of websites that embed trackers in the *Before-Accept* and *After-Accept* visits. Visiting from outside Europe leads to an increase of Tracking on the *Before-Accept* in all cases, while, on the *After-Accept*, changes are limited.

3.5 Impact on Complexity and Performance on Top-100k Websites

In this section, we measure the impact of accepting privacy policies on the webpage characteristics and loading performance. Trackers and Third-Party objects that the browser has to load and display upon consent may impact the amount of data to download and the rendering performance. Here, we do not restrict on a per-country

or per-category analysis and use the crawl on the top-100,000 websites according to the Tranco global list.

For each website, we visit only the landing page, doing a *Warm-up* visit to fill the browser cache, followed by a *Before-Accept* and *After-Accept*. We compare results on the latter two visits, considering only those websites for which *Priv-Accept* successfully accepted the privacy policy, which happens on 23% of websites. This is in line with the previous findings, as the Tranco list is a worldwide rank and includes (i) European websites in a language different from those for which we built the keyword list and (ii) websites based in non-European countries for which regulations do not apply. To give more insights, we detail the percentage of websites with a Consent Banner on the Tranco list in Figure 3.18a, computed every 5,000 websites in the rank. The solid red line reports the percentage for websites popular in the 5 European countries we target. Websites belong to this set if (i) they appear in the Similarweb ranks for the 5 countries or (ii) the Top-Level Domain belongs to the 5 countries.[14] Out of these 6,917 websites, *Priv-Accept* accepts the privacy policy on 3,861 (55.8%), which is close to what we have obtained with the Similarweb ranks (54.7%). This percentage does not change with website popularity. Conversely, for the remaining websites (blue dashed line), the share of websites where *Priv-Accept* found a Consent Banner is 32% for the top-ranked and then it settles around 20%, hinting that some globally popular websites tend to implement a Consent Banner even if they are based outside Europe, using a language supported by *Priv-Accept* (likely English). In 2020, Hills *et al.* [34] found that popular CMPs are present on almost 10% of websites in the top-10 k Tranco list. Here, with *Priv-Accept*, we can affirm that Consent Banners (regardless the employed CMP) appear in more than 30% for the same set of websites.

The high number of Consent Banners found for the 5 European countries reflects in a large increase of the number of Third-Parties from the *Before-Accept* to the *After-Accept*, as shown in Figure 3.18b. The solid red line highlights that these websites already include, on average, 11.1 Third-Parties in the *Before-Accept*. In the *After-Accept*, the average grows to 17.3. Differently, the increase for the non-EU websites is smaller – see the area between the blue solid and dashed lines. In the *Before-Accept*, Third-Parties are larger than for the 5 European countries if we compare the solid blue and red lines. This is due to the larger presence of non-EU websites,

[14]The Tranco list does not provide a per-country rank.

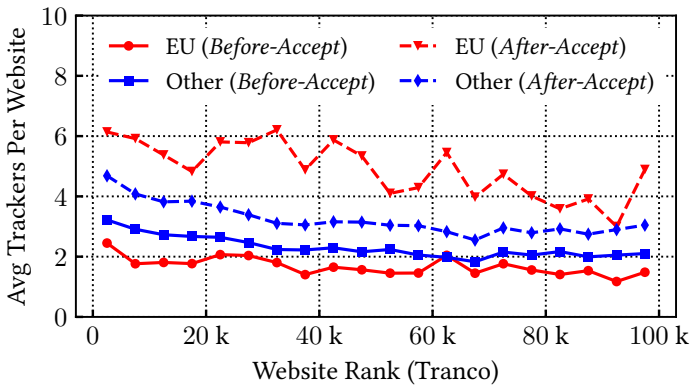


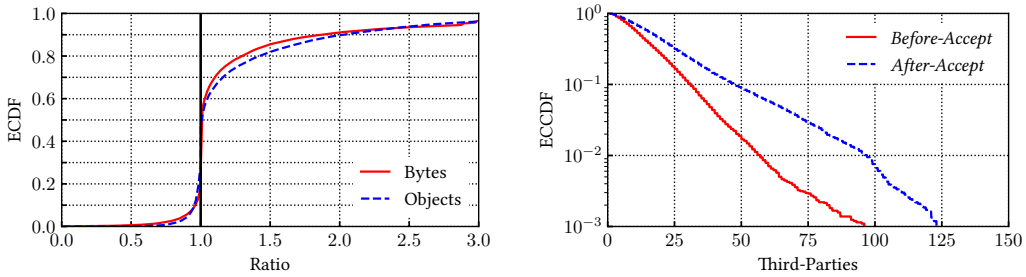
Fig. 3.19 Average number of Trackers per website (Tranco list).

which do not have to implement a Consent Banner. In the *After-Accept* (dashed blue line), the increase is moderate, not reaching the values of the 5 European countries (red dashed line), potentially because *Priv-Accept* misses many *Accept-button* in non-supported languages and of possible custom tracking domains not present in our lists. FigureFor the sake of completeness, Figure 3.19 shows the number of Trackers instead of Third-Parties, providing similar insights.

3.5.1 Impact on Page Objects and Size

We focus on the webpage complexity in terms of bytes and objects to download. We compute the ratio R between the measurement on the *Before-Accept* and *After-Accept*, i.e., $R = x_{After}/x_{Before}$, where x is the metric of interest. We show the results in Figure 3.20a, separately for total downloaded bytes and objects. As expected, accepting the privacy policy increases the webpage size ($R > 1$) by a sizeable factor. For instance, about 9% of websites download more than twice the objects, and about 5% of websites sees an increase of 3 times or more.

Interestingly, we also observe some websites that are lighter in the *After-Accept* than in the *Before-Accept*. Investigating further, these cases are mostly due to the lack of additional content upon acceptance coupled with the saving of not loading the CMP objects on the *After-Accept*. This happens commonly on those websites that either add a Consent Banner despite not using tracking mechanisms, or that contact Trackers and Third-Parties even before the user has accepted the privacy



(a) Distribution of the page size (in bytes and objects) ratio over all websites.

(b) Distribution of the number of Third Parties. Notice the log scales.

Fig. 3.20 Webpage characteristic before and upon consent to privacy policies (Tranco list).

policies. While the former might be seen as an excess of caution, the latter cases are likely violating the privacy regulations.

To better characterize the differences, we quantify the number of Third-Parties seen in the *Before-Accept* and *After-Accept*. We show the Empirical Complementary Cumulative Distribution Function (ECCDF) in Figure 3.20b. On median, websites rely on 12 Third-Parties on the *Before-Accept*. This figure grows to 17 on the *After-Accept*. The ECCDF highlights the tail of the distribution where we observe those websites that rely on a very large number of Third-Parties: the percentage of websites with more than 50 grows from 1.8% to 9.2%, with 3.0% including more than 75 Third-Parties upon acceptance. This growth in the number of Third-Parties is mostly due to an increase of Trackers and objects related to advertisements that gets loaded after accepting the privacy policy. We also perform statistical tests to compare whether the mean and median of the two sample distributions are statistically different at level 0.05 (t-Test for the mean and Mood's test for the median). Both result statistically significant in *After-Accept*.

Plotting the number of Trackers instead of Third-Parties, leading to similar conclusions. We show it in Figure 3.21.

3.5.2 Impact on Page Load Time

The Third-Party domains appearing after acceptance are generally devoted to advertisements, analytics and Web tracking. Contacting them has direct implications on the page load time and, indirectly, on the users' QoE [108]. We thus expect this

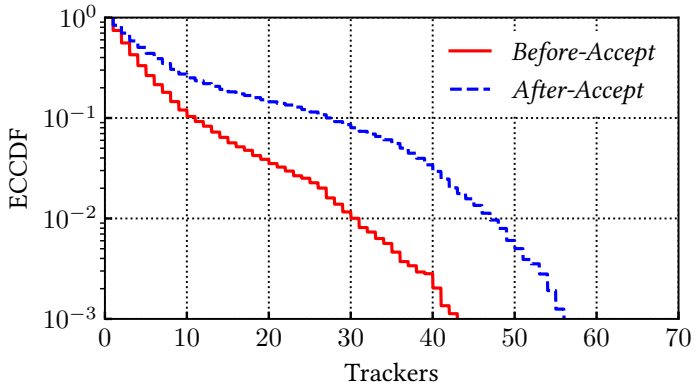
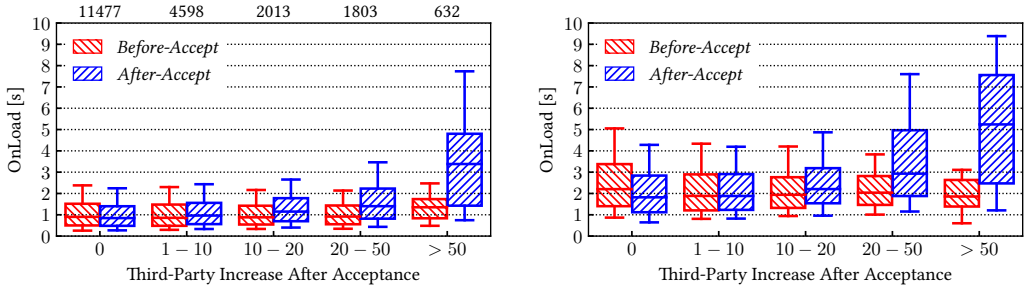


Fig. 3.21 Distribution of the number of Trackers (Tranco list).



(a) Warm Browser Cache.

(b) Cold Browser Cache.

Fig. 3.22 OnLoad time of websites versus the increase of Third-Party number upon acceptance (Tranco list).

to cause an increase on the page load time because the browser has to resolve the server name via DNS and contact more servers. For instance, this ultimately limits the advantages offered by new protocols like the stream multiplexing and the header compression offered by HTTP/2 and HTTP/3.

To gauge this, we dissect the webpage load time in Figure 3.22, comparing separately visits with a warm cache (Figure 3.22a) and with a cold cache (Figure 3.22b). The cardinality of each category is reported on the top axis of the left-most figure. In case of warm cache, we run a *Warm-up visit*, then the *Before-Accept* and *After-Accept*. In case of of cold cache, we run the *Before-Accept* without a *Warm-up visit*. Then we erase the HTTP cache and socket pool, then we run the *After-Accept*.

We report the distributions of the *onLoad* time for websites with similar number of additional Third-Parties that are loaded in the *After-Accept*. We use boxplots,

where the boxes span from the first to the third quartile and whiskers from the 10^{th} to 90^{th} percentile. The central stroke represents the median. The number of websites in each set is detailed on the top the respective boxplot. As expected, the more Third-Parties are loaded upon acceptance, the larger the time needed to load the webpage and the larger its variability. Especially for the websites that add more than 10 Third-Parties, the distributions are remarkably different on the *Before-Accept* and *After-Accept*. Considering visits with cold browser cache (Figure 3.22a), those website with 20–50 additional Third-Parties, the median *onLoad* time passes from 0.91 to 1.41 seconds. The difference increases for the 632 websites adding more than 50 Third-Parties upon acceptance. Here, the median *onLoad* time increases from 1.35 to 3.38 seconds, more than doubling. Notice also the tail of 25% of websites loading in more than 4.8 seconds, which happens in less than 2% of cases during the *Before-Accept*. We already observed such an increase in our previous study [57], where we measured that median *onLoad* time increases by 1.3*s* when cookies policies are accepted. We statistically compare all these couples of sample distributions between *Before-Accept* and *After-Accept*, testing differences in the median at a significance level 0.05 (Mood's test). The test is passed in all cases, showing statistically significant differences.

Similar considerations hold for visits with a cold browser cache (Figure 3.22b). As expected, with the clean cache, websites load time increases – compare values in figs 3.22a and 3.22b. Those that do not add new Third-Parties tend to load slightly faster on the *After-Accept*, potentially due to the absence of the Consent Banner. In fact, differences are statistically significant in the median of the distributions between *Before-Accept* and *After-Accept*, except for the group 1–10 additional Third-Parties. Again, we observe that those adding several Third-Parties after acceptance have much higher *onLoad* time on the *After-Accept* than on the *Before-Accept*: The median *onLoad* time increases from 1.8 to 5.2 seconds. Finally, we observe that the *onLoad* time values tend to be lower than what measured in older works, potentially because of the advances of content delivery network and increased hardware and software performance. Bocchi *et al.* [62] measured a median *onLoad* time of 3s in 2016 on a similar albeit smaller set of websites.

Chapter 4

The Topics API

As discussed in Chapter 2.2, Google has introduced the Topics API as a new proposal to replace the third-party cookies framework. Given the role of its proponent in the Web ecosystem, Topics API has the potential to affect a large portion of Web users and websites. An analysis of its privacy properties is thus needed. In the following, we will introduce the Topics API, how they work, what risks do they carry for the privacy of the users — which they aim at preserving.

This chapter is mostly based on a conference paper presented at the *2023 Privacy Enhancing Technologies Symposium* [15], and an extended version currently under revision at the time of writing [119].

4.1 The Topics API

In this section, we describe how the Topics API operates for creating a profile from the user's browsing history. We consider a browser that a user employs to navigate the Internet.[1] We assume time is divided into epochs of duration ΔT (one week in the current proposed Topics API operation). During each epoch e, the browser collects and counts the number of visits to each website and forms a *bag of websites* $\mathscr{B}_{u,e}$ for the user u. It keeps track only of the website hostnames the user *intentionally* visited, e.g., by typing its URL, or by clicking on a link in a web page or other applications. Formally, given a user u and the epoch e,

[1] We intentionally confuse the terms *user* and *browser* to identify the person and the application they use to navigate the Internet.

Table 4.1 Main terminology to model Topics API algorithm and threat model.

Symbol	Definition
n_{topic}	Number of topics in the taxonomy
E	Number of past epochs included in the profile
p	Probability a random topic to replace a real topic
N	Epochs of observation by the attacker
U	User population set
$\lambda_{u,t}$	Rate of visit by user u to topic t
$\mathscr{B}_{u,e}$	Bag of visited *websites* by user u at epoch e
$\mathscr{T}_{u,e}$	Bag of visited *topics* by user u at epoch e
$\mathscr{P}_{u,e}$	Profile for the user u at epoch e
$\mathscr{P}_{u,e,w}$	Exposed Profile to website w for user u at epoch e
$\mathscr{G}_{u,N,w}$	Global Reconstructed Profile by w after N epochs
$\mathscr{R}^{f}_{u,N,w}$	Denoised Reconstructed Profile by w after N epochs with threshold f

let $\mathscr{B}_{u,e} = \{(w_1, f_{1,u,e}), (w_2, f_{2,u,e}), \ldots, (w_n, f_{n,u,e})\}$, where w_i represent the visited websites and $f_{i,u,e}$ the number of times u visited w_i during epoch e.

The Topics API algorithm operates in the browser and processes the history of $\mathscr{B}_{u,e}$ over the past E epochs to create a corresponding *Exposed Profile* $\mathscr{P}_{u,e,w}$ for the user u, epoch e and each specific website w the user visits during the current epoch. In fact, the browser builds a separate *Exposed Profile* for each visited website w to mitigate re-identification attacks. We base the following description on the public documentation of the Topics API available online.[2] The operation of the Topics API has the following steps.

Step 1 - From websites to topics For each of the websites $w_i \in \mathscr{B}_{u,e}$, the browser extracts a corresponding *topic* t_i. To this end, the browser uses a Machine Learning (ML) classifier model that returns the topic of a website given the characters and strings that compose the website hostname. At this step, each browsing history $\mathscr{B}_{u,e}$ is transformed into a *topic history* $\mathscr{T}_{u,e} = \{(t_1, f'_{1,u,e}), (t_2, f'_{2,u,e}), \ldots, (t_m, f'_{m,u,e})\}$ where t_i represents the topic the model outputs, and $f'_{i,u,e}$ counts its total occurrences. Each website is mapped to a topic and the original frequencies $f_{i,u,e}$ are summed by topics into $f'_{j,u,e}$. There are n_{topic} which form a taxonomy of possible interests

[2] https://developer.chrome.com/docs/privacy-sandbox/topics/, accessed on Monday 22nd January, 2024

the users have. Such taxonomy will include between a few hundred and a few thousand topics (the IAB Audience Taxonomy contains about 1,500 topics)[3]. In our experiments, we employ the Google ML model implemented in Chrome. In its first implementation, it supports $n_{topic} = 349$ topics[4] and the model is based on a Neural Network trained by Google using a manually curated set of 10,000 domains.[5] It leverages website hostnames only and neglects any other part of a URL.[6]

Step 2 - From Topics to Profiles Given the topic history $\mathscr{T}_{u,e}$ for user u at epoch e, the browser selects the z most frequently visited topics and stores them into the *profile history* $\mathscr{P}_{u,e}$, which will be referred as the user u Profile at epoch e in the following. If the topic history $\mathscr{T}_{u,e}$ contains less than z topics for a user u in epoch e, the Topics API adds to the Profile $\mathscr{P}_{u,e}$ padding, random topics from the taxonomy until z topics are included. z is currently set to 5.

Step 3 - Per-website topic selection The first time the user visits the website w, the browser generates a *Exposed Profile* $\mathscr{P}_{u,e,w}$. For each past epoch $i \in \{e-1,\ldots,e-E\}$, the browser selects at random one topic t_i^* from the profile history $\mathscr{P}_{u,i}$. To increase privacy guarantees, with probability p the browser replaces the topic t_i^* with a random topic t_{rnd} uniformly selected from the global topic list. p is currently suggested to be 0.05. $\mathscr{P}_{u,e,w}$ contains thus at most E topics (a topic picked from $\mathscr{P}_{u,e-1}$, a topic from $\mathscr{P}_{u,e-2}$, etc.). Once generated, the Exposed Profile remains the same for the whole epoch e.

Usage by websites From this point on, each time the user visits the website w during the current epoch, the website w may request the browser to share the current Exposed Profile $\mathscr{P}_{u,e,w}$ and use the returned topics to provide behavioural advertising. Notice that the Exposed Profile $\mathscr{P}_{u,e,w}$ is built only for websites intentionally (first-party) visited by the user u. Any third-party service (e.g., a component embedded on

[3] https://iabtechlab.com/standards/audience-taxonomy/, accessed on Monday 22nd January, 2024

[4] https://github.com/patcg-individual-drafts/topics/blob/main/taxonomy_v1.md, accessed on Monday 22nd January, 2024

[5] Google announced a second version of the taxonomy (https://developer.chrome.com/blog/topics-enhancements/). However, at the moment of writing, Google still has not released the code to map websites to this second set of topics.

[6] The mapping from a website to a category is prone to inaccuracies and depends on the employed ML model. Here we do not consider the implications of such errors. See https://developer.chrome.com/docs/privacy-sandbox/topics/#classifier-model, accessed on Monday 22nd January, 2024

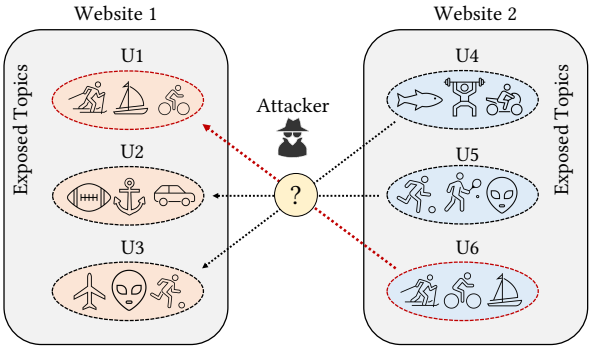


Fig. 4.1 Threat model sketch: An attacker leverages the Exposed Profiles obtained from the Topics API to re-identify the same user in the population of two websites.

the webpage of site w, but hosted on a different domain) will receive topics of the first-party websites w it is embedded into. That is, all trackers embedded into the website w receive always the Exposed Profiles $\mathscr{P}_{u,e,w}$ of w.

Periodic Profile update At the beginning of the epoch $e+1$, the browser computes the new profile history $\mathscr{P}_{u,e+1}$ and discards $\mathscr{P}_{u,e-E}$. Similarly, if and when the user visits again the website w, the browser creates $\mathscr{P}_{u,e+1,w}$ from $\mathscr{P}_{u,e,w}$: it includes a new topic selected from $\mathscr{P}_{u,e+1}$ (Step 3), and removes the oldest topic, i.e., the one originally belonging to $\mathscr{P}_{u,e-E+1}$ (keeping the others). This means that a website continuously visited by a user can observe up to one new topic per epoch (and such topic may be randomly extracted).

4.2 Attacks against the Topics API

4.2.1 Threat model

In this thesis, we consider the threat model introduced by the same proponents of Topics API [89, 91] and discussed in a technical report by Mozilla [90]. In detail, we consider the risk of re-identification — i.e., the possibility to link a *Reconstructed User Profile* from an audience to a known individual; or that two websites use the profiles to match people within their audiences. Such possibility has already been

evaluated in the literature on similar contexts [85–87, 15]. We sketch the threat model in Figure 4.1.

As in [89, 91], we assume that a website w uses first-party cookies to track a user over time so that it can reconstruct the set of topics users in its audience are interested in. Then, it matches the reconstructed profiles with the target profile of the victim (or with all profiles of the second website audience).

In this threat model, the attacker accumulates the Exposed Profiles $\mathscr{P}_{u,e,w}$ over epochs, overcoming the limitation introduced by Topics API to limit the Exposed Profiles to at most one new topic per epoch, for at most E epochs. Let us assume w observes its users $u \in U(w)$ for N epochs (i.e., epochs in $[1,N]$). At the end of the process, for each user u, w builds the *Global Reconstructed User Profile* $\mathscr{G}_{u,N,w}$, where $\mathscr{G}_{u,N,w} = \cup_{e\in[1,N]}\mathscr{P}_{u,e,w}$. [7] In the long run, the set of topics could act as an identifier string (or fingerprint) for user u, enabling the re-identification process either with the set of topics of a known user or with users from the audience U_2 of website w_2.

Notice that this attack may be carried out by a third-party service s. In this case, we assume some websites w_1 and w_2 collude with s. Both w_1 and w_2 embed s. They both share with s the user identifier each time a user visits them. The third party then builds $\mathscr{G}_{u,N,w_1}$ and $\mathscr{G}_{u,N,w_2}$ autonomously so that it can match the profiles of users in both audiences.[8]

4.2.2 Random and Rare Topic Denoising

The Global Reconstructed Profile $\mathscr{G}_{u,N,w_i}$ is noisy and unstable, as it is built directly on the set of exposed topics. Indeed, some topics might be observed only once on website w_1 and never on w_2, or vice versa. This could happen with i) random topics used as replacements by the API (Step 3 of Section **??**), ii) rare topics that

[7] Please note that by observing the exposed topics for N epochs, the attacker actually accumulates $N+E-1$ observations (E in the first epoch, one in the others). We opted to simplify the notation by assuming one topic per epoch, so that N epochs correspond to N topics observed.

[8] Notice not every third-party s will receive a topic. Only if s observed the user visit a site w about the topic in question within the past E weeks, then s is allowed to receive such a topic (see https://github.com/patcg-individual-drafts/topics). We ignore such limitation, i.e., we assume that the third party s is pervasive enough to make this condition irrelevant because the third party is present on the most popular websites, which will enable the reception of every topic. This is the case with popular web trackers.

seldom appear in the profile history $\mathscr{P}_{u,e}$ and thus are not consistently exposed to both websites, iii) padding topics that fill the profile history $\mathscr{P}_{u,e}$ in case a user has visited less than z topics in an epoch (Step 2 of Section **??**). To prevent these topics from hindering re-identification, the attacker uses filtering mechanisms to obtain a *Denoised Reconstructed Profile* $\mathscr{R}^{f}_{u,N,w_i}$, where f is a threshold: the set $\mathscr{R}^{f}_{u,N,w_i}$ contains only those topics that appear in at least f different weeks.

Preliminary studies on the Topics API, such as [90], mainly considered the need for identifying the random topics in an Exposed Profile $\mathscr{P}_{u,e,w}$ by carrying out a simple statistical test based on the number of times a topic is exposed by a user. The author of [90] states that observing a topic more than once is sufficient to infer its authenticity with high confidence. We further extend our previous work [15] which considered a moving threshold by discussing the effect of single threshold values.

In this work, we consider a filtering threshold f, with the goal of filtering not only random topics but any rare topics that might impair re-identification. We evaluate different values of f, including $f = 1$ (no threshold). Intuitively, f should increase with larger N, as rare topics have a greater chance of appearing multiple times. The probability a given topic is exposed as a random topic is p/T. Thus, the probability it is included in a profile with threshold f at epoch N as:

$$p_{above}(f,N,p,T) = 1 - \sum_{k=0}^{f-1} \binom{N}{k} \left(\frac{p}{T}\right)^{k} \left(1 - \frac{p}{T}\right)^{N-k}.$$

With $p = 0.05$, $T = 349$, $N = 30$, and $f = 2$, the probability of a random topic t being included in a profile is in the order of 10^{-8}. Here, our goal is not only to exclude random topics but also to filter out real-but-rare topics. In Section **??**, we show that this filtering is essential to achieve attack effectiveness.

4.2.3 The attacks

In this thesis, we consider two attacks, the *Strict* and the *Loose* attacks. We consider two websites w_1 and w_2 with populations U_1 and U_2, with $|U_1| = |U_2| = 1{,}000$. By construction, we include the same persona v (the victim) to both U_1 and U_2. We then evaluate the probability of re-identifying v in w_1 and w_2.

Strict Attack

In the *Strict* Attack, v is matched to v' iff the following two conditions occur:

- w_1 and w_2 reconstruct the same Denoised Reconstructed Profile i.e., $\mathscr{R}^f_{v,N,w_1} = \mathscr{R}^f_{v',N,w_2}$.
- The Denoised Reconstructed Profile $\mathscr{R}^f_{v,N,w_1}$ is unique in U_1, and $\mathscr{R}^f_{v',N,w_2}$ is unique in U_2.

Let

$$P_E := Prob\left(\mathscr{R}^f_{v,N,w_1} = \mathscr{R}^f_{v,N,w_2}\right),$$

where P_E is the probability that v exposes the same Denoised Reconstructed profile on both sites. Let

$$P_U := Prob\left(\mathscr{R}^f_{v,N,w_1} \text{ unique in both } U_1 \text{ and } U_2\right).$$

Note that, by construction, denoted with:

$$P_U^{(2|1)} := Prob\left(\mathscr{R}^f_{v,N,w_1} \text{ unique in } U_2 \mid \mathscr{R}^f_{v,N,w_1} \text{ unique } U_1\right)$$

and

$$P_U^{(1)} := Prob\left(\mathscr{R}^f_{v,N,w_1} \text{ unique in } U_1\right),$$

we have $P_U^{(2|1)} \cdot P_U = P_U^{(1)}$.

Thus, the probability of *correct* re-identification, i.e., a True Positive (TP), can be computed as:

$$Prob(correct\ re\text{-}identification) = P_E \cdot P_U = P_E \cdot P_U^{(1)} \cdot P_U^{(2|1)}$$

Similarly, let

$$\begin{aligned}\overline{P_E} =& Prob\Big(\exists!\, v' \in U_2, \text{ with } v' \neq v : \mathscr{R}^f_{v,N,w_1} = \mathscr{R}^f_{v',N,w_2}, \\ & \mid \mathscr{R}^f_{v,N,w_1} \text{ unique in } U_1\Big).\end{aligned}$$

$\overline{P_E}$ is the conditional probability of *incorrect* re-identification on the event $\{\mathscr{R}^f_{v,N,w_1}$ is unique in $U_1\}$.

The probability of an incorrect re-identification, i.e., a False Positive (FP), becomes:

$$Prob(incorrect\ re\text{-}identification) = P_U^{(1)} \cdot \overline{P_E}.$$

In other words, given a match between two unique profiles $v \in U_1$ and $v' \in U_2$, the re-identification is successful and correct, i.e., a TP, if $v' = v$. If instead $v' \neq v$, the re-identification is successful but wrong, i.e., a FP.

Loose Attack

With respect to the *Strict* Attack, the *Loose* Attack adopts a different matching rule: the attacker matches v and v' if:

- The Denoised Reconstructed Profile on w_1 is a subset of the Global Reconstructe Profile on w_2; and viceversa, i.e., $\mathscr{R}^f_{v,N,w_1} \subseteq \mathscr{G}_{v',N,w_2}$ and $\mathscr{R}^f_{v',N,w_2} \subseteq \mathscr{G}_{v,N,w_1}$.
- No other user v'' exists such that $\mathscr{R}^f_{v,N,w_1} \subseteq \mathscr{G}_{v'',N,w_2}$ and $\mathscr{R}^f_{v'',N,w_2} \subseteq \mathscr{G}_{v,N,w_1}$.

As in the *Strict* Attack, we can compute the probability of a user being re-identified as follows:

$$Prob(correct\ re\text{-}identification) = \widehat{P}_U \cdot P_S,$$

where

$$P_S := Prob(\{\mathscr{R}^f_{v,N,w_1} \subseteq \mathscr{G}_{v,N,w_2}\} \cap \{\mathscr{R}^f_{v,N,w_2} \subseteq \mathscr{G}_{v,N,w_1}\})$$

and

$$\widehat{P}_U := Prob(\cap_{v' \neq v}(\{\mathscr{R}^f_{v,N,w_1} \not\subseteq \mathscr{G}_{v',N,w_2}\} \cup \{\mathscr{R}^f_{v',N,w_2} \not\subseteq \mathscr{G}_{v,N,w_1})\}) \qquad (4.1)$$

while

$$Prob(incorrect\ re\text{-}identification) = Prob(\exists! v' \in U_2, \text{ with } v' \neq v : (\{\mathscr{R}^f_{v,N,w_1} \subseteq \mathscr{G}_{v',N,w_2}\} \cup \{\mathscr{R}^f_{v',N,w_2} \subseteq \mathscr{G}_{v,N,w_1}\})). \tag{4.2}$$

Intuitively, the *Loose* Attack allows more flexibility in matching the same user on different websites. For example, a user could expose a topic a different number of times on two different websites, causing the threshold f to filter it in one of them. In the *Strict* Attack, this would cause the user not being re-identifiable, while the *Loose* Attack, taking into consideration both the Denoised Reconstructed Profile $\mathscr{R}^f_{u,N,w}$ and the Global Reconstructed Profile $\mathscr{G}_{u,N,w}$, would be able to identify the profiles as belonging to the same user.

On the other side, this flexibility comes with an increase in the number of false positive matches.

Asymmetric Weighted Hamming Attack

For comparison, we consdier the Asymmetric Weighted Hamming Attack (AWHA) introduced in [91]. Authors of [91] analytically prove offers optimal accuracy. — although only under some specific assumptions.While leaving the details to the original work, here we just present the main feature of the AWHA:

- For every user having visited website w_1, the attacker computes the *sequence* of exposed topics, keeping the temporal dimension.
- Among all the users having visited w_2, the attacker chooses the one that maximizes the similarity of two profiles by minimizing the weighted Hamming distance of the two sequences.
- When comparing two users' sequences, a weighted element-wise distance is evaluated considering whether the two users have exposed the same topic in epoch e, or not. The total distance between two users is the sum of the element-wise distances.
- The user in w_2 with the smallest weighted distance from the user in w_1 is matched.

The AWHA always chooses a user $u_1 \in U_1$ to match user $u_2 \in U_2$. Contrary to the *Strict* Attack and the *Loose* Attack, the AWHA algorithm always returns a match, largely increasing the false positives as we will discuss in Chapter 4.5.1.

4.3 Dataset

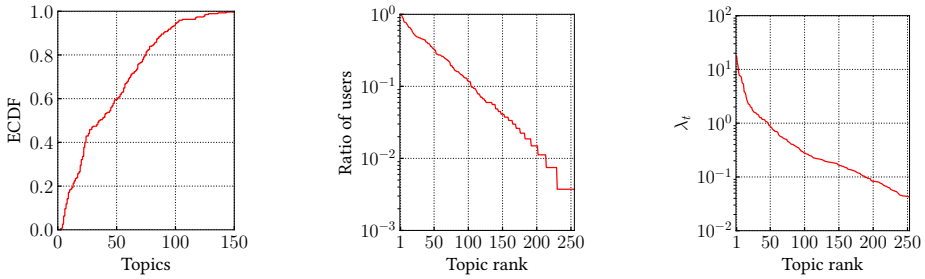
To simulate the Topic API algorithm in a realistic environment, we rely on a dataset of real browsing histories collected from a population of users who joined a Personal Information Management System (PIMS).

4.3.1 Data collection methodology

In the context of the PIMCity project[9], we designed, implemented and deployed a fully-fledged online PIMS called EasyPIMS and opened it for experimentation [18]. A PIMS (Private Information Management System) is a framework which offers users the possibility to upload their data and control over the purposes and the ways their data are used. Using EasyPIMS, a simple web interface allows the users to provide fine-grained consent for sharing the data with data buyers and eventually to monetise their data in a marketplace. Among various types of data, the platform allows users to share their browsing history by installing a browser plugin for Google Chrome or Microsoft Edge on their PCs running any operating system. Such plugin records all *intentionally visited webpages* and stores them in a central repository. During the test of our PIMS, we recruited 3,369 volunteers who had the possibility of using the platform for four months in 2022. Out of them 928 installed the plugin. To join the PIMS, there was no restriction on the geographic area, and users belong to 35 different countries in Europe, Asia, and America. Considering the demographic information of the population, 478 are male, 226 are female and 224 did not declare their gender. The age ranges from 18 to 72 years, the average being 33.

In this chapter, we leverage the actual browsing histories of EasyPIMS users who explicitly provided their consent for research purposes to the usage of their browsing history and any personal data we use. 613 gave such permissions. Among those, we restrict the population to those users that actively used the platform. Since the Topics

[9] https://www.pimcity-h2020.eu/, accessed on Monday 22nd January, 2024



(a) Number of topics per user on the entire observation period.

(b) Ratio of users visiting a topic. Topics in decreasing popularity.

(c) Average rate of visits per topic. Topics in decreasing rate.

Fig. 4.2 Characterization of topic visits.

API operates on a weekly basis, we consider a user to be active in a given week if they visited at least 10 webpages. In total, we obtain 267 users that were active in at least one week. We use the sequence of websites visited by these users for our study.

Ethical Aspects Our data collection process is compliant with ethical principles and EU privacy regulations. EasyPIMS was part of a European Project involving 12 partners and the European Commission has approved all the data collection and processing procedures. Users voluntarily participated, were informed, explicitly opted-in via the PIMS web interface, and were rewarded by sweepstakes. We only use data of users who explicitly provided their consent for research, which the user has to select explicitly. Moreover, data processing has been carried out in an anonymous fashion using a secure computing infrastructure running up-to-date software and with restricted physical access to authorized personnel. During data processing, we only process data regarding browsing histories, neglecting all other attributes, such as name, gender, or geographic location.

4.3.2 Characterization of users and topics

In total, our dataset includes $2,813,283$ webpage visits to $50,976$ different websites. The number of visits per user per week varies significantly, with some users using the platform for a few weeks and others for the whole four-month experimental period. Some users even installed the plugin on multiple browsers and devices (e.g., desktop and laptop PCs), increasing the amount of data collected by their account. In median, active users access 222 web pages each week, with 26.1% of users that

visit less than 50 pages; conversely, 14% of the users visit more than $1,000$ pages. Considering unique websites a user visits in a week, in median, active users access 30 different websites, while the 25^{th} and 75^{th} percentiles of the distribution are 10 and 71 websites, respectively.

Using the current implementation of the Topic API ML model Google opened since Chrome 101, for each of the $50,976$ websites w in our dataset, we extract the corresponding topic t the API returns. We obtain 250 topics visited at least once by a user in our dataset. In the following, we report the characterization of the topic visits.

Focus first on the number of unique topics each user visited at least once during the entire experimentation. This is useful to understand how complicated (and unique) could be a Profile $\mathscr{P}_{u,e}$. We report the ECDF in Figure 4.2a. The distribution is quite spread: in the median users visit 36 topics, with the most diverse users visiting more than 150 topics. Conversely, a handful of users visit less than 5 topics. Not reported here for the sake of brevity, the median number of topics each user visits per week is 17, with a maximum of about 70. Only less than 10% of users visit less than 5 topics in some weeks.

Figure 4.2b reports the ratio of users visiting a given topic. The top 5 topics are Search Engines, News, Arts & Entertainment, Internet & Telecom, and Business & Industrial. The most popular topic is visited by 99,3% of users, while up to 100 (200) topics are visited by at least 10% (1%) of the users.

At last, we show the average rate of visits per topic in Figure 4.2c. We compute first the rate of visits of user u to topic t $\lambda_{u,t} = \sum_e f'_{t,u,e}/T$, being T the total activity time (discretized by weeks) of user u in the whole observation window. Then, we compute the average rate of visits among the subset $U_{|t}$ of users that visited the topic t as

$$\lambda_t = \sum_{u \in U_{|t}} \frac{\lambda_{u,t}}{|U_{|t}|} \tag{4.3}$$

Notice a topic that is globally unpopular can still sizeably appear in the Profile of those few users frequently visiting such topic. In fact, the construction of the topic history $\mathscr{T}_{u,e}$ depends on the rate of visits $\lambda_{u,t}$ the user u has for the topics t she/he is interested in during the e-th epoch. Our dataset allows us to estimate $\lambda_{u,t}$ for all users.

Overall, we believe these figures reflect the natural variability of users. Despite being limited, our dataset includes a real population of users browsing the web, with different interests, backgrounds, nationalities, etc. Unfortunately, we cannot advocate our dataset is representative of general human behaviour and we do not exclude it may be biased in some direction such as gender or education. We use it to study the impact of the Topic API algorithm to prevent an attacker from mounting a re-identification attack.

In the following, we present two models that allow us to generate some possible realistic population U and to study the probability two websites can link the profile of the same user.

4.4 Population models

We consider two models for the generation of the users U that extend and generalize a mere trace-driven approach that replicates the browsing pattern of each user in our dataset. The models allow us to generate an artificial population U of any desired size $|U|$: the first model generates personas with the same first-order statistical properties of the users in the trace; the second model combines the visiting rates of the users in our dataset.

Real Users We consider each of the 268 users in the dataset. A user is characterized by a list of visit rates $\lambda_{u,t}$ for all $t = 1, 2, \ldots, n_{topic}$. $\lambda_{u,t}$ is calculated by averaging the occurrences $f'_{t,u,e}$ along the period in which the user u has been active in our collection system. $\lambda_{u,t} = 0$ if the given user never visited topic t.

I.I.D. Personas We create a population of independent and identically distributed (i.i.d.) personas obeying the same marginal statistics as the set of real users from our dataset. To this end, we leverage (i) the marginal ECDF of the number of topics per user (Figure 4.2a), (ii) the marginal empirical distribution of the topic popularity (Figure 4.2b), and (iii) the average empirical rate of visits for each topic λ_t (Figure 4.2c). In such a way, we can create a population of any size $|U|$ that shares the same first-order statistical properties as the population of our dataset. We adopt the inverse transform sampling method [120] for the generation of the random

variable that follows a known ECDF. In detail, we generate a persona u according to a three-step process:

1. We extract the number of topics c_u the persona is interested in from the empirical marginal distribution of the number of topics per user (Figure 4.2a).
2. We choose the set of the topics $C_u = \{t_i\},\ i = 1,2,\ldots,c_u$ by extracting with no repetitions c_u topics from the normalized version of the empirical distribution of the topic popularity (Figure 4.2b).
3. For each $t \in C_u$, we assign an effective visit rate λ_t from Equation 4.3, which equals the average empirical visiting rate (Figure 4.2c).

Notice that in step 2 we select each topic essentially independently (just disregarding possible repetitions). This breaks existing correlations among topics and may appear in part unrealistic. In fact, it is known that real users show highly-correlated interests which reflects in highly-correlated topics [121]. The resulting personas in U have instead all the same statistical properties, increasing the probability of having similar profiles. As such this model is a rather pessimistic scenario for the attacker.

Crossover Personas We generate each persona u according to the biologically-inspired crossover procedure during the generation of offspring. We start the process from the population U^* of Real Users. We then randomly select two parent individuals p_0 and p_1 from U^* and generate a new persona u. It inherits part of the genome (i.e., visit rates to topics) from p_0 and part from p_1. For this, we generate a binary mask and assign the rate of p_0 (p_1) if the corresponding bit is true (false). In this third model, the correlation of the appearance of topics is stronger than in the previous case. For this, we expect this scenario to be optimistic for the attacker since the uniqueness of personas is boosted by making them more heterogeneous and easier to re-identify.

4.4.1 Simulation of visits and profile creation

Given the population U, we assume each persona u visits topic t according to a homogeneous Poisson process with the assigned rate $\lambda_{u,t}$. At each epoch e, for each topic t and persona u, we thus extract a Poisson-distributed random variable that

represents the number of visits user u performs to t. This allows us to obtain the topic history $\mathscr{T}_{u,e}$, and from it the Profile history $\mathscr{P}_{u,e}$ which contains only the top-z topics (Step 2 of Topic API algorithm). Next, we generate the Exposed Profile $\mathscr{P}_{u,e,w}$, possibly offering w a random topic instead of a real top topic (Step 3).

By repeating the periodic profile update procedure at the beginning of each epoch $e+1$, we simulate the process for N epochs so that, at the end, w fills the Denoised Reconstructed Profile $\mathscr{R}_{u,N,w}$ for each persona $u \in U$ after filtering the Global Reconstructed Profile $\mathscr{G}_{u,N,w}$.

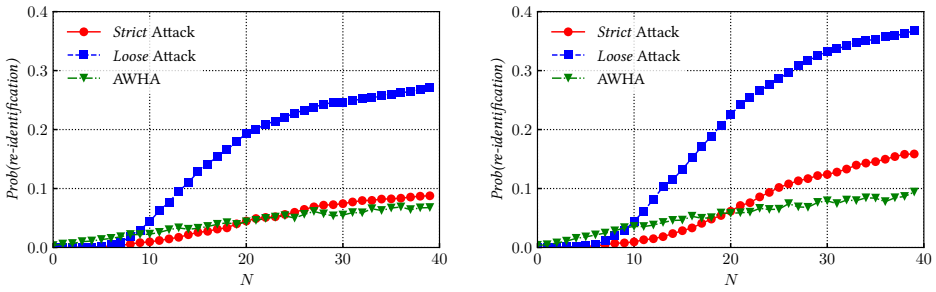
4.5 Results

In this section, we illustrate the results of our study. We first compare the effectiveness of the different attacks presented in Chapter 4.2. Then, we evaluate how the probability of a user being re-identified changes according to the denoising threshold chosen and the number of users in the system.

In the following, where not expressly otherwise stated, we set the denoising threshold $f = 2$ and consider the Google suggested values for the Topic API parameters ($z = 5,\ E = 3,\ p = 0.05, \Delta T = 1$ week). We repeat each experiment 10 times and report the average performance. As introduced in Chapter 4.2.1, we consider two websites w_1 and w_2 aiming at re-identifying a user based on the topics that each website has observed. As a reference metric, we consider the ratio of users that each attack correctly matches between two websites, and define it as *Prob(re-identification)*. Similarly, we define *Prob(incorrect re-identification)* as the ratio of incorrect matches.

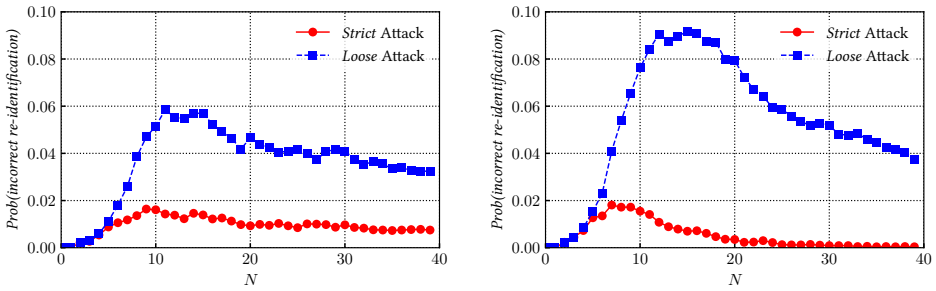
4.5.1 Comparison of attack models

We first compare the performance of the three attacks presented in Chapter 4.2, and show the results in Figure 4.3, where the x-axis represents different epochs and the y-axis the reidentification probability *Prob(re-identification)*. As expected, increasing the number of epochs, all attacks become more effective and the *Prob(re-identification)* increases. Overall, the *Loose* Attack (blue line) shows to have up to $4\times$ better performance with respect to the *Strict* Attack (red line) and the AWHA



(a) I.I.D. personas. (b) Crossover personas.

Fig. 4.3 Probability of a user being correctly re-identified across the epochs, by the means of different attacks.



(a) I.I.D. personas. (b) Crossover personas.

Fig. 4.4 Probability of a user being incorrectly matched across the epochs, with the *Strict* Attack and *Loose* Attack.

(green line), reaching around 25% in *Prob(re-identification)* after $N = 30$ epochs and almost 28% after $N = 40$ epochs, for I.I.D. personas (Figure 4.3a) and almost 38% for Crossover personas (Figure 4.3b). With Crossover personas *Prob(re-identification)* moderately improves, as personas are, by construction, more heterogeneous. As mentioned in Chapter 4.2, the *Loose* Attack has a larger flexibility than the *Strict* Attack, allowing the attacker to account for behaviours that can affect the performance of the *Strict* Attack: e.g., a topic overcoming the denoising threshold on website w_1, while not being able to do so in w_2. The other two attacks achieve worse performances, below 10% (20% with Crossover personas). The likelihood-based AWHA proposed by [91], although the better *Prob(re-identification)* in the very first epochs, grows slower with time than the other two attacks.

While Figure 4.3 shows the probability an attacker *correctly* identifies the same user on w_1 and w_2, an incorrect re-identification may happen. In other words, the

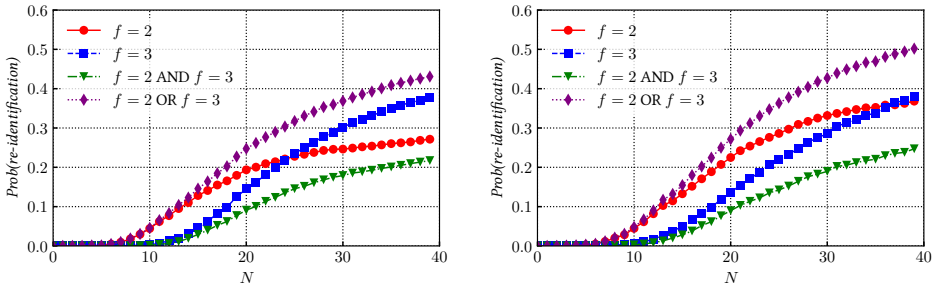
attack provides a match for the target user, but with an incorrect user from the other website. We show the probability of this event (i.e., the *Prob(incorrect re-identification)*) for the *Strict* Attack and the *Loose* Attack in Figure 4.4. Since the AWHA attack always matches a user's profile with the most likely profile on the other website, the rate of users incorrectly matched is complementary to the number of users correctly matched. This does not happen in the *Strict* Attack and *Loose* Attack, where the attack matches no profile if the conditions are not met. Thus, the ratio of incorrectly matched users for AWHA would largely outnumber the one of *Strict* Attack and *Loose* Attack, and therefore we choose not to display them in Figure 4.4.

Both the *Strict* Attack and the *Loose* Attack show an increase in the *Prob(incorrect re-identification)* in the first epochs, peaking between $N = 5$ and $N = 15$. In this phase, users' profiles are still very similar one to the other, causing more users to be incorrectly matched. Increasing the epochs, the attacker builds a richer (and thus more unique) profile and improves the re-identification chances: after $N = 30$ epochs, with I.I.D. personas, the error rate is around 4%, while the *Prob(re-identification)* increases above 20%. For the *Strict* Attack, the *Prob(incorrect re-identification)* never exceeds 2%, converging toward 0% with Crossover personas and 1% with I.I.D. personas. This confirms that the *Strict* Attack is more conservative than *Loose* Attack in providing a match, but those matches are more accurate. In summary, with enough time, the *Strict* Attack and especially the *Loose* Attack are efficient enough to provide an interesting option for an attacker. On the other side, recall that the AWHA outputs too many false matches for an attack to be valuable.

In the remainder of this section and in Chapter 4.6, we will only consider the *Loose* Attack, as it outperforms the other two attacks. We keep comparing the results with both I.I.D. and Crossover personas.

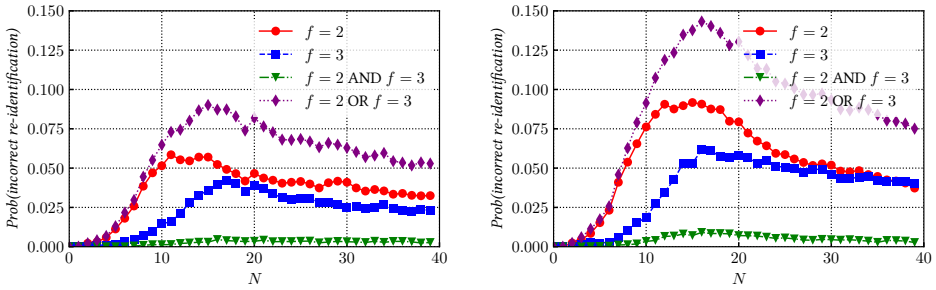
4.5.2 Impact of the denoising filter

In this section, we discuss the impact of the attacker choice for the denoising threshold f. We expect that imposing no threshold (i.e., $f = 1$) leads to almost null performance, and, to maximize effectiveness, the attacker should set f to 2 or 3. They could even consider combining the results obtained by using both the thresholds. Notice that f should increase with epochs N as the attacker has a higher



(a) I.I.D. personas. (b) Crossover personas.

Fig. 4.5 Probability of a user being correctly re-identified across the epochs, by the means of different threshold rules.



(a) I.I.D. personas. (b) Crossover personas.

Fig. 4.6 Probability of a user being incorrectly re-identified across the epochs, by the means of different threshold rules.

probability of observing multiple times the same random or rare topics. This was already evident in Figure 4.3b: The *Prob(re-identification)* for the *Loose* Attack Attack flattens when N exceeds 30. This is in great part caused by setting $f = 2$, which becomes less effective the more epochs the attacker observes topics exposed by users.

To better understand the impact of f, we show in Figure 4.5 how *Prob(re-identification)* evolves with $f = 2$ and 3. Later, we also propose a couple of compound strategies. For the sake of readability, we omit to represent the case with $f = 1$: in fact, the *Prob(re-identification)* never exceeds 3% for both population models demonstrating that a filtering strategy is necessary to achieve attack effectiveness. Let us first focus on the curves representing the *Prob(re-identification)* with $f = 2$ and $f = 3$. Using $f = 2$ (red line), the attacker re-identifies users earlier, because, in a few epochs, new topics populate $\mathscr{R}$. However, when the number of epochs increases, the attack becomes less effective, allowing a number of random and rare topics to pollute $\mathscr{R}$. Indeed, those topics make the reconstructed profile of a given user different on the two websites, thus impeding re-identification. At that point, the attacker shall increase the threshold to $f = 3$, which can better cope with the larger magnitude of noise introduced by rare and random topics. When $f = 3$ (blue curve), the attack is less effective in the first epochs — since too few topics exceed the threshold resulting in an (almost) empty profile $\mathscr{R}$. Conversely, it performs better when the number of epochs becomes sufficiently large. Setting $f = 3$ outperforms $f = 2$ when $N > 24$ and $N > 36$ for I.I.D. and Crossover personas, respectively.

Combining strategies

Now, we consider two additional strategies that combine the sets of the users re-identified with threshold $f = 2$ and $f = 3$ to make a final decision. In the first strategy, the attacker considers a user to be re-identified if the user appears *in both sets*; this represents a conservative approach. In the second strategy, the attacker considers a user re-identified if they appear *in at least one of the sets*; this represents a daring approach.

It is important to clear a possible misunderstanding at once: one could consider, for instance, that the set of users re-identified with $f = 2$ *and* $f = 3$ is the same as the set of users re-identified with $f = 3$, thinking that if a user is re-identified with $f = 2$, than they will be re-identified also with the stricter threshold $f = 3$. However,

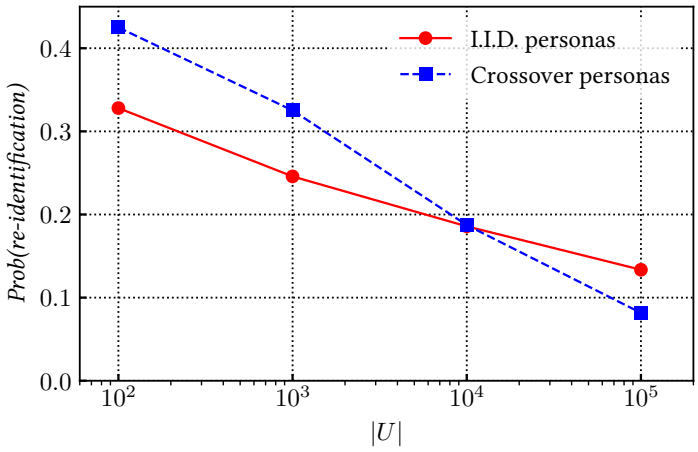


Fig. 4.7 Probability of being re-identified with different numbers of personas.

due to the filtering threshold a user's profile can be unique [10] with $f = 2$ but not with $f = 3$, causing the user to be re-identified in one case but not the other.

These two filtering strategies work as lower and upper bounds when tuning the trade-off between the fraction of re-identified users and the error rate. With the first approach (green curve, labelled as "$f = 2$ AND $f = 3$" in Figure 4.5), *Prob(re-identification)* is always below the $f = 2$ and $f = 3$ cases. Conversely, with the second approach (purple curve, labelled as "$f = 2$ OR $f = 3$'), *Prob(re-identification)* is always higher. Different is the picture for the error rate — the *Prob(incorrect re-identification)*— depicted in Figure 4.6. The cautious attacker that uses the AND approach obtains a negligible *Prob(incorrect re-identification)*, thus maximizing the high correct/incorrect match ratio. An attacker willing to maximize the *Prob(re-identification)* would instead opt for the OR approach, which, however, leads to a sizeable *Prob(incorrect re-identification)*. Also in terms of *Prob(incorrect re-identification)*, the two classical attacks with $f = 2$ and $f = 3$ stand in the middle as expected.

4.5.3 Impact of the number of users

We now fix $f = 2$, $N = 30$ and vary the number of users $|U|$. Intuitively, the number of users in the set of candidates has an impact on the probability of a user being

[10] With abuse of language, under the *Loose* Attack, we say that users's profile is unique if the event in r.h.s. of (4.1) occurs.

re-identified. The larger the website's audience, the harder the reidentification is. We illustrate this effect in Figure 4.7, where we show how reidentification probability varies when increasing the number of users in the audience of w_1 and w_2 — notice the log scale on the x-axis. In a larger pool of users, there is a higher probability of finding another user exposing a similar combination of topics. This makes the user identical to more than one individual in the eyes of the attacker, thus preventing re-identification. Recall that *Strict* Attack and *Loose* Attack do not make any guess if a user does not have a unique Denoised Reconstructed Profile. Notice, however, that the decrease of the *Prob(correct re-identification)* slows down with a larger number of users $|U|$ both with I.I.D. and Crossover personas, following a logarithmic decrease: even with a pool of 10^5 users, the *Prob(re-identification)* is not negligible. Moreover, also consider that other techniques (such as browser fingerprinting) could be used by an attacker to reduce the set of possible reidentification candidates.

4.6 The role of Topics API design parameters

In this section, we study the impact of the Topics API parameters on the *Prob(re-identification)*. In particular, we investigate the roles of z, i.e., the number of topics that are selected every epoch to build the profile $P_{u,e}$ of a user, and p, i.e., the probability at which an exposed topic is replaced with a random one. In the following experiments, we consider $N = 30$, $|U| = 1,000$, $f = 2$.

4.6.1 The number of topics in the profile

In Figure 4.8, we show how the choice of the parameter z impacts the probability of a user being correctly re-identified for I.I.D. personas (red curve) and Crossover personas (blue curve), in a scenario with $1,000$ users. When exposing a limited number of topics (in the extreme case, only the top topic from the previous week), the *Prob(re-identification)* decreases because of the low informative value of the top topic(s) (e.g., `Search Engine`), which are popular among most users and do not characterize a specific individual. Interestingly, the *Prob(re-identification)* hits a maximum with $z = 3,4$, depending on which personas model we consider. This is the best setting for the attacker: the available combinations of exposed topics differentiate users, meaning they are easier to be linked and thus re-identified. Further increasing

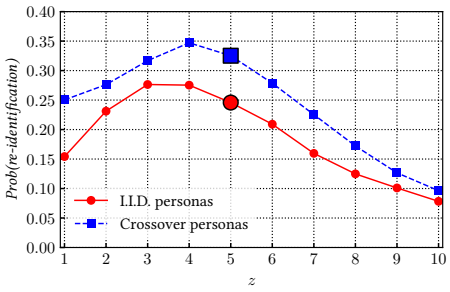


Fig. 4.8 The probability of an attacker correctly re-identifying a user, with different values of z. $N = 30$, $|U| = 1{,}000$, $f = 2$, $p = 0.05$. We highlight the *Prob(re-identification)* with the default value $z = 5$.

z rapidly impairs the *Prob(re-identification)*, which goes towards zero. This is caused by the padding introduced by the Topics API when the number of exposed topics in a week by a user is smaller than z, which happens with increasing probability with larger z. The random topics added as padding have two consequences:

- Every week, many users' profiles $\mathscr{P}_{u,e}$ are filled with random topics, generated independently every week. This breaks the stationarity assumption that benefits the *Loose* Attack (as well as the *Strict* Attack) since the users' behaviour over time becomes unpredictable.
- Even if all the z topics are real (i.e., really belong to the user and are not injected randomly), a larger pool to choose from slows down the convergence of the reconstructed profiles by both websites. A website collects, at each epoch, one topic. Thus, the larger the z, the larger the number of epochs needed to collect them all.

4.6.2 The role of random topics

We now set $f = 2$, $N = 30$, $|U| = 1{,}000$, $p = 0.05$ and we quantify the impact of the probability of exposing a random topic p on the attack effectiveness. In Figure 4.9, we show how *Prob(re-identification)* varies with different values of p. Notice that $p = 0.05$ corresponds to the current default value of the Topics API. Increasing p has a negative effect on the probability of re-identifying a user. This is no surprise, as a larger p increases the probability of replacing a real with a random topic. Recall

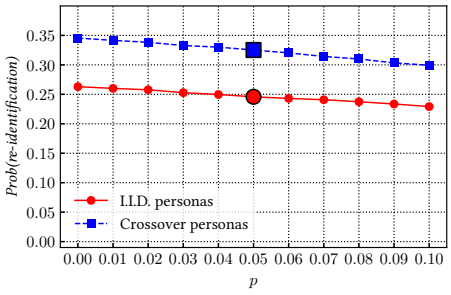


Fig. 4.9 Probability of a user being re-identified, with different values of p. $N = 30$, $|U| = 1,000$, $f = 2$, $z = 5$. We highlight the *Prob(re-identification)* with the default value $p = 0.05$.

the topic replacement takes place independently for each website, thus making the reconstructed profiles different.

Interestingly, the introduction of the random topic does not significantly impact the *Prob(re-identification)*. In fact, it decreases by less than 5% between $p = 0.0$ and $p = 0.10$. This is due to the effectiveness of the filtering threshold that can remove the random topics quite efficiently. An interesting line of future work would be to evaluate the trade-off between the improvements in the privacy guarantees introduced by p and the impact on the data utility (from the advertiser's perspective) caused by the introduction of false information.

Chapter 5

The z-anonymity

On the Web and beyond, there is the need to learn how to manage data that are continuously generated by several different systems. Privacy-Preserving Data Publishing (PPDP) techniques are useful to manipulate such data while respecting the privacy of the users to whom the data belong. In particular, we focus on continuous, streaming data that should be anonymized on the fly (e.g., the websites visited by a large amount of users). As discussed in Chapter 2.3, many algorithms focus on the streaming data anonymization scenario, but few of them tackle the zero-delay goal between data reception and its anonymized version publication. z-anonymity is one of these, and in this Chapter we discuss its privacy properties. We discuss what the z-anonymity is, and what privacy feature it supports, particularly in relation with k-anonymity.

The content of this chapter is mainly based on an article published on *Performance Evaluation* [17], which in turn is an extension of a previous conference paper presented at *2020 IEEE Big Data* [16].

5.1 z-anonymity: anonymization for data streams

5.1.1 The z-anonymity property

We suppose to operate with data streams, where we continuously receive observations that associate users with attributes. We define an observation as a tuple (t,u,a),

indicating that, at time t, the user u exposes the attribute a from a catalog of attributes $\mathscr{A}$. Attributes can be related to whatever field: a visit to a web page, a purchase, a GPS location, etc.

Here, we assume every attribute $a \in \mathscr{A}$ is a *quasi-identifier*. That is, in the stream there are no *sensitive attributes* – i.e., attributes that contain private information, but cannot bring to re-identification of the user.

The users are completely described by the set of quasi identifiers $\mathscr{A}$.

We want to keep private those values of attributes associated with small groups of users, which could ease the re-identification. As presented in [122], we define the property of z-private attribute as follows:

Definition 1. An attribute a is z-private at time t if it is exposed by less than z users in the past Δt time interval.

Notice that the same attribute a can be both z-private and not z-private for different t. If the anonymized dataset hides all z-private attributes, it achieves z-anon.

Definition 2. A stream of observations is z-anonymized if it does not contain z-private attributes at any t, given z and Δt.

In other words, the attributes that are associated with less than z users in the past Δt shall be removed or replaced with an empty identifier. The goal is to prevent rare attributes from being published, thus reducing the possibility of an attacker to re-identify a user. In the following, we show how it is possible to achieve z-anon in real time efficiently.

5.1.2 Implementation and complexity

Our goal is to design an algorithm to achieve z-anon which satisfies the following requirements:

- **Zero delay:** the anonymization property should be achieved without introducing a delay in publishing the anonymized stream. In other words, we want to make an atomic decision. All approaches based on the processing of batches of observations are not applicable, as they need to store and process the entire batch before the release.

- **Efficient algorithm for high dimensional data:** the anonymization property shall be achieved with an efficient algorithm, allowing the deployment at high speed and large volume of data with off-the-shelf computing capabilities. It is important to carefully build an algorithm working with efficient data structures to obtain the necessary information as quickly as possible. Moreover, users might expose a large set of attributes, whose number is not known *a priori*.

The algorithm we propose generalizes the approach presented in a previous work [122]: the attributes a are stored as a hash table $\mathscr{H}$, with linked lists to manage collisions. Each value $\mathscr{H}(a)$ in the hash table contains three elements:

- metadata about a;
- a Least Recently Used list LRU_a of tuples (t,u);
- a hash table $\mathscr{V}_a$ to track the users that exposed a.

The idea is to minimize the time spent searching into the data structures, therefore reducing the memory accesses. By assuming that the number of attributes a is one order of magnitude smaller than the hash structure dimension, collisions are infrequent, and consequently, the total computational cost is $O(1)$ for each incoming observation.

The $\mathscr{H}(a)$'s metadata include the counter c_a and the reference for the LRU_a first and last attribute. Referring to Algorithm 1, once an observation (t,u,a) arrives, the value a should be inserted in the hash table, if not already present (lines 2-6), otherwise an update should be performed (lines 7-16). The hash value is calculated and the access to the table is done in $O(1)$.

If the user u exposes an attribute a for the first time in the previous Δt, the user u is inserted into $\mathscr{V}_a$ in $O(1)$, c_a is increased by one and the tuple (t,u) is inserted on top of the LRU_a in $O(1)$ thanks to the aforementioned references (lines 8-11). If u was already present in $\mathscr{V}_a$ and in LRU_a with value (t',u), we replace t' with t and the tuple (t,u) is moved on the top of the LRU_a. Again all is done in $O(1)$ (lines 12-14).

Last, to evict old entries and consequently decrease c_a, we traverse the LRU in reverse order: we remove each tuple (t',u') where $t' < t - \Delta t$, and we decrease c_a accordingly (lines 18-22). At last, if $c_a \geq z$ the observation $(t, f(t,u), a)$ is released (lines 23-24). The $f(t,u)$ needs an explanation: every Δt, users' identifiers are

Algorithm 1 Pseudo code of the algorithm to achieve z-anon.

Input: (t,u,a)
if $a \notin \mathscr{H}$ **then**
    $\mathscr{H} \leftarrow \mathscr{H} \cup a$ *//new attribute: insert it for the first time*
    $\mathscr{V}_a \leftarrow \{u\}$ *//insert new user u*
    $LRU_a \leftarrow (t,u)$
    $c_a = 1$
else
    if $u \notin \mathscr{V}_a$ **then**
        $\mathscr{V}_a \leftarrow \mathscr{V}_a \cup \{u\}$ *//insert new user u*
        $c_a \leftarrow c_a + 1$ *//add new user*
        $LRU_a \leftarrow (t,u)$
    else
        $(t',u) \leftarrow (t,u)$ *//update timestamp of user u*
        move (t,u) on top of LRU_a
    end if
end if
//Always evict old users
for $((t',u')$ = last(LRU_a); $t' < t - \Delta t$; (t',u')=next) **do**
    remove (t',u') from LRU_a
    remove (u') from $\mathscr{V}_a$
    $c_a \leftarrow c_a - 1$
end for
if $(c_a \geq z)$ **then**
    OUTPUT $(t, f(u,t), a)$
end if

rotated, such that the ID related to a user u at a time t_0 will no more be related to u at $t_0 + \Delta t$. The user identifiers thus depend on the time at which the tuple is published; the attacker will not be able to track the behaviour of the same user after a Δt.

Notice that k-anon has been proved [123] to be an *NP-Hard* problem. Differently, z-anon property can be achieved for each observation with $O(1)$ complexity with properly sized hash-tables. Implementation proposed in [122] allows to manage in real time a 40 Gbit/s stream with common hardware. As part of the PIMCity project, we provide also a Python library to let interested users to adopt z-anon.[1].

We exemplify the algorithm to enforce z-anon in Figure 5.1. Assume $z = 3$. At time t_0 user u_0 is the first to expose the attribute a_0. The attribute a_0 is z-private at time t_0, hence it shall be obfuscated. Still, the information that u_0 exposed the attribute a_0 shows its effects for a time equal to Δt. At time t_1, user u_1 also exposes a_0.

[1] https://pypi.org/project/zanon/ — accessed on Monday 22nd January, 2024.

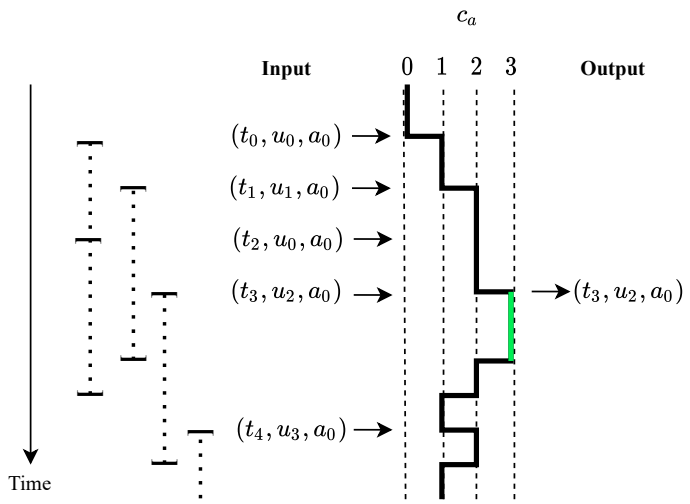


Fig. 5.1 A graphical example of z-anon concept with $z = 3$: a tuple is released only if at least other $z - 1 = 2$ different users have exposed the same attribute in the previous Δt.

Since the number of observations in Δt is still smaller than 3, this observation is not released. At time t_2 user u_0 re-exposes a_0, extending the lifetime of the observation, but not changing the number of unique users having exposed a_0. At time t_3, user u_2 exposes a_0, making the total users in the past Δt equal to 3. Thus the attribute a_0 is not z-private at time t_3 and the observation (t_3, u_2, a_0) can be be released. At time $t_1 + \Delta t$ the attribute a_0 related to user u_1 expires, hence the total user count decreases back to 2. The same happens when u_0 observation expires (at $t_2 + \Delta t$), so that when u_3 exposes a_0 at t_4 the observation can no more be released.

z and Δt are parameters that can be tuned to achieve the desired trade-off between data utility and privacy. Therefore, z-anon can be adapted to the needs of the desired use case. A large z and a small Δt result in the majority of attributes to be anonymized, while a small z and a large Δt allow rare values to be possibly released.

The Δt parameter can be set based on how often the system administrator randomizes the identifiers of users – such that a user u is no more related to the same identifier after a time period Δt: its choice may depend on several aspects, such as the nature of the application, the input stream rate, the system memory (the larger Δt, the larger the memory requirement to store the users' information), and so on.

Notice that z-anon considers individual attributes, not on their combinations, as for the k-anon property. Hence, it is interesting to study which guarantees the z-anon algorithm offers in a global perspective, i.e., which guarantees it is possible to give

Table 5.1 Terminology used to model z-anon and k-anon.

$\mathscr{U}, U$	Set and number of users
$\mathscr{A}, A$	Set and number of attributes
Δt	The time interval length used for evaluating z-anon
λ_a	Exposing rate for attribute a
R_a	Random variable counting number of times a user exposes attribute a in Δt. $R_a \sim Poisson(\lambda_a \cdot \Delta t)$
X_a	Random variable representing whether a user exposes attribute a in Δt. $X_a \sim Bernoulli(p_a^X)$
O_a	Random variable representing whether a tuple (t,u,a) is published when exposed. $O_a \sim Bernoulli(p_a^O)$
Y_a	Random variable representing whether a user published at least once attribute a in Δt. $Y_a \sim Bernoulli(p_a^Y)$
$\overline{Y}$	Set of random variables $\{Y_a\}_{a \in \mathscr{A}}$. $\overline{Y} \sim Bernoulli(p_{\overline{y}})$
$Q_{\overline{y}}$	Random variable representing the number of users $u \in \mathscr{U}$ with the same realization $\overline{y}$ in ΔT. $Q_{\overline{y}} \sim Binomial(U-1, p_{\overline{y}})$
p_{k-anon}	Probability that a realization of $\overline{Y}$ satisfies k-anon property

on the privacy properties in terms of k-anon of the output. We study this relationship between z-anon and k-anon in Chapter 5.2.

5.1.3 Modeling z-anonymity

To fully understand the effect of applying z-anon, we model the input data stream as a stochastic process and we show how anonymization modifies it. We release the code implementing the model.[2] Table 5.1 summarizes the terminology we use throughout the following sections.

Modeling the data stream

We consider a system in which a set of $\mathscr{U}$ users can access the catalog $\mathscr{A}$ of attributes. Let $U = |\mathscr{U}|$ and $A = |\mathscr{A}|$. Users generate a stream of information, exposing in real-time the attribute they have just accessed. The system collects *tuples* in the form (t,u,a), i.e., at time t, the user $u \in \mathscr{U}$ exposes the attribute $a \in \mathscr{A}$.

For now, we assume that users are homogeneous and generate independent tuples, so that the probability of exposing a specific tuple depends only on a. We will relax this assumption by considering classes of users in Chapter 5.4. We assume any user

[2] https://github.com/nikhiljha95/zanonymity — accessed on Monday 22nd January, 2024.

Table 5.2 The default values used for the model.

Variable	Default Value
U	1 000
A	20
λ_{a_r}	0.2 / r
z	150
k	2
Δt	12

u exposes the attribute a according to a homogeneous Poisson process with rate λ_a. Hence, the number of times a user exposes the attribute a in a time period Δt is modeled as a Poisson distributed random variable R_a with parameter $\lambda_a \cdot \Delta t$, i.e., $R_a \sim Poisson(\lambda_a \cdot \Delta t)$.

In our analyses, we assume a small set of popular attributes and a large tail of infrequent ones. This allows us to represent systems where users are more likely to expose top-ranked attributes, but there exist a large catalog, a condition which is often observed in real-world systems that are governed by power-law distributions [124]. The usability of the model does not depend on these assumption, which are just considered to match several real-world scenarios. As such, we choose that the λ_a for all attributes follows a power law in function of their rank. Let us suppose attributes are sorted by rank, where the most popular attribute is a_1 and the least popular a_A. In the implementations we will show, we impose $\lambda_{a_1} = 0.2$ and set the remaining λ_a as the power-law function $\lambda_{a_r} = {}^{0.2}/_{r}$, where r is the rank of attribute a_r. The default parameters used in this article are collected in Table 5.2.[3]

We denote as X_a the random variable describing whether a user exposed at least once the attribute a in a time interval Δt. X_a assumes value 1 if the user exposes a in Δt, 0 otherwise. We note that, by construction, $X_a \sim Bernoulli(p_a^X)$, where p_a^X denotes the probability that a user has exposed attribute a, at least once, in the past Δt. It is straightforward to compute p_a^X given λ_a and Δt as:

$$p_a^X = P[R_a \geq 1] = 1 - P[R_a = 0] = 1 - \exp(-\lambda_a \cdot \Delta t) \tag{5.1}$$

Notice that the different attributes are independent and p_a^X is not a distribution probability mass function, hence the sum of p_a^X over $a \in \mathscr{A}$ can be different from 1.

[3]We change the default parameter values from [16] to limit the computational complexity induced by the new model. See 5.2.2 for a discussion of model scalability.

Applying z-anon

We show how a stream of data modeled as above appears after being z-anonymized. Under z-anon, z-private attributes at time t are not released. Here, we define the indicator random variable O_a associated to the event that the exposed tuple (t,u,a) is published, whose probability of occurring is denoted with p_a^O. (t,u,a) is published if a is not z-private at time t.

$$p_a^O = P[O_a = 1] = P\left[\sum_{v \in \mathscr{U} \setminus u} X_a \geq z - 1\right] \tag{5.2}$$

Given our assumption of independence and homogeneity across the users, we are summing up $U-1$ independent and identically distributed random variables, which are distributed as X_a. Note that we exclude user u, since we are checking the z-anon conditionally over the tuple (t,u,a). Hence the current user is already involved by construction.

Since X_a is a Bernoulli with success probability p_a^X, its sum, which is Binomially distributed, counts the number of occurrences in a sequence of $U-1$ independent experiments, $\sum_{v \in \mathscr{U} \setminus u} X_a \sim Binomial(U-1, p_a^X)$.

Starting from Equation 5.2 and using the probability mass function of the Binomial distribution we can derive p_a^O as:

$$p_a^O = 1 - \sum_{i=0}^{z-2} \binom{U-1}{i} \left(p_a^X\right)^i \left(1 - p_a^X\right)^{U-1-i} \tag{5.3}$$

Similarly to Equation 5.1, we denote as Y_a the random variable describing whether a user published at least once the attribute a in a time interval Δt. Again, $Y_a \sim Bernoulli(p_a^Y)$, where p_a^Y is simply:

$$p_a^Y = P[X_a = 1] \cdot P[O_a = 1] = p_a^X \cdot p_a^O$$

The set of random variables describing the presence or absence for all the possible attributes $a \in \mathscr{A}$ for a user is denoted as $\overline{Y} = \{Y_a\}_{a \in \mathscr{A}}$. The attacker will not know the random variable $\overline{Y}$, and will observe only realizations of it. Let us denote as y_a a

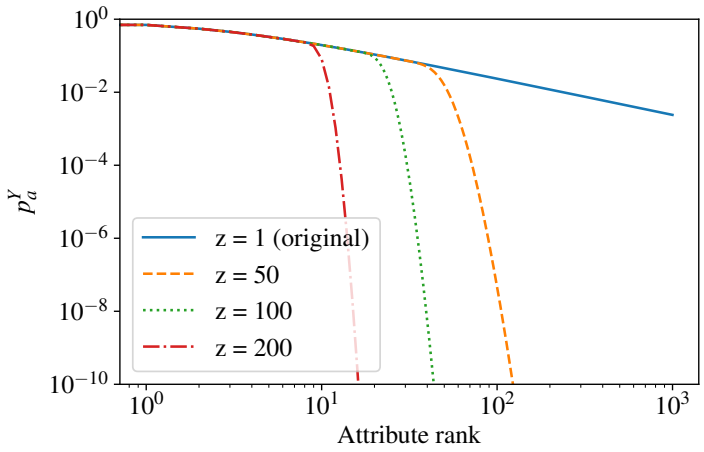


Fig. 5.2 The probability p_a^Y for a user to publish attribute a in Δt.

realization of the random variable Y_a and as $\overline{y} = \{y_a\}_{a \in \mathscr{A}}$ a realization of the random variable $\overline{Y}$.

Impact on released data

We now qualitatively show the effect of applying z-anon on the released data stream. In this experiment, we set $A = 1\,000$ and $U = 1\,000$. We suppose the popularity of attributes follows a power law, with $\lambda_{a_r} = 0.2/r$, where r is the attribute rank.

We study the probability of observing the attribute a in a Δt, for a given user, in both the original and released data. Figure 5.2 shows p_a^Y in function of the attribute rank. The blue solid line represents the probability of observing an attribute in case $z = 1$, i.e., no anonymization ($p_a^Y = p_a^X$). The curve appears as a straight line, representing a power law on the log-log plot. When enabling z-anon ($z > 1$), we notice that the probability of observing uncommon attributes abruptly decreases with an evident knee. For example, if we observe the curve for $z = 100$ (green dashed line in the figure), already the 13^{th}-ranked attribute is released with a probability below 10^{-6}, while it appears on the original stream with probability 10^{-1}. A higher z moves the knee of the curve closer to the top-ranked attributes.

Conversely, increasing the number of users U increases the lowest-ranked attributes probability to be published (with more users in the system, it is easier to

satisfy the z threshold). This would move to the right the curves' knee. We discuss the impact of U later in Chapter 5.3.4.

In summary, the figure shows how z-anon prevents uncommon attributes from being released. Indeed, those attributes are released only when enough users are exposing them, hence only for popular attributes. In the following, we propose a probabilistic model to study how a z-anonymized data stream can result in a k-anonymized dataset with controllable probability.

5.2 Modeling k-anonymity

We now study the relationship between the z-anon and k-anon properties. Intuitively, z-anon ensures that each published value of an attribute a has been exposed at least by z users in the past time interval, while, with k-anon, any given record (i.e., the combinations of all user's attributes) must appear in the published data at least k times. With high-dimensional data, the set of attribute combinations becomes extremely large, thus making k-anon tricky to guarantee. Here we show that, with a proper choice of z, it is possible to release data in which user results k-anonymized.

5.2.1 Getting to k-anon

Given a specific realization $\bar{y}$ of a user, our goal is to derive the probability to observe at least other $k-1$ users in $\mathscr{U}$ having the same realization $\bar{y}$. If this happens, the user is k-anonymized.

Recall that we assume attributes to be independent. Thus each realization $\bar{y} = \{y_a\}_{a\in\mathscr{A}}$ happens with a probability $p_{\bar{y}}$, which results to be:

$$p_{\bar{y}} = \prod_{a\in\mathscr{A}} \left[y_a \cdot p_a^Y + (1-y_a)\cdot(1-p_a^Y) \right]. \tag{5.4}$$

For any realization $\bar{y}$, the random variable representing the number of users with the same realization in the users' set (which we can model as $Q_{\bar{y}}$) is described by a Binomial distribution with parameters $U-1$ and $p_{\bar{y}}$: $Q_{\bar{y}} \sim Binomial(U-1, p_{\bar{y}})$.

From the point of view of an external observer which only accesses the privatized stream, a user has thus a probability of being k-anonymized which can be retrieved from the law of the total probabilities:

$$p_{k-anon} = \sum_{\bar{y}} \left[1 - \sum_{j=0}^{k-2} P\left[Q_{\bar{y}} = j\right] \right] \cdot p_{\bar{y}}. \tag{5.5}$$

In Equation 5.5, the probability for a user of finding at least $k-1$ other users with the same $\bar{y}$ is evaluated as the opposite of finding up to $k-2$ users. Then, we average this quantity over all the possible realizations of the random variable $\overline{Y}$, summing over all the $\bar{y}$ and multiplying by the respective $p_{\bar{y}}$ to obtain the final p_{k-anon}.

In summary, our model describes the probability that a data stream undergoing z-anon results in a dataset which respects the k-anon property. Although we can only provide probabilistic guarantees on the k-anonymization of the released data, our model allows one to study and control this probability as a function of the parameters.

Moreover, note that, even with no z-anon in place (i.e., $z = 1$), the model provides a general way to evaluate the probability of a data stream being k-anonymized in a transaction dataset with U users and a catalog of A attributes.

An analysis on the model results as compared to simulation ones is provided in Chapters 5.3.1 and 5.3.5.

5.2.2 Model approximation

As it emerges from Equation 5.5, the evaluation of p_{k-anon} depends on the number of possible realizations $\bar{y}$ of $\overline{Y}$. In a configuration with A binary attributes, there are 2^A of such possible realizations, and their enumeration represents a computational bottleneck. In the following, we propose an approximation strategy to make the computation of Equation 5.5 practical. We introduce two parameters, θ_1 and θ_2. The first operates to limit the number of attributes to consider. The second limits the number of realizations to evaluate. The rationale is that many realizations have usually a negligible probability to happen, allowing us to neglect them while keeping unchanged the model accuracy.

Effective attributes

Focus first on θ_1. Here we leverage the typically heavy-tailed nature of attribute popularity. Intuitively, the least-popular ones will be so rare that no realization $\bar{y}$ would contain them. z-anon will exacerbate this, since it will further decrease the publications of such unpopular attributes.

Let the *effective attributes* be those we expect to be exposed at least by θ_1 user in ΔT. Let $A_{eff} \leq A$ be their number. Considering the expected value, we have that an attribute a is effective if $\mathbb{E}[U \cdot p_a^Y] \geq \theta_1$, from which $p_a^Y \geq \theta_1/U$. We can filter those attributes for which $p_a^Y < \theta_1/U$. In a nutshell, we discard all realizations where non-*effective attributes* appear and, thus, reduce their number from 2^A to $2^{A_{eff}}$.

Effective realizations

Even the realizations derived from the $2^{A_{eff}}$ attributes may not all be worth an evaluation in Equation 5.5. We thus design an algorithm to reduce the number of realizations to consider, by enumerating the most likely ones and discarding the rarest ones. To this end, we organize all the possible realizations in a tree. Let the root realization be $\bar{y}_0$. We show a toy example in Figure 5.3, for three effective attributes ($A_{eff} = 3$), and use it as a running example. The root node ($\bar{y}_0$) holds the most probable realization. Hence, $\bar{y}_0 = \{y_a\}_{a \in \mathscr{A}}$, where:

$$y_a = \begin{cases} 1, & \text{if } p_a^Y \geq 0.5 \\ 0, & \text{otherwise.} \end{cases} \tag{5.6}$$

In our example, $p_a^Y < 0.5, \forall a$, and the most probable realization is $[0,0,0]$. $\bar{y}_0$ has three child nodes, each obtained by changing a single attribute. We arrange the children from the most probable to the least probable. The probability of these realizations depends on the distance of $p_{a'}^Y$ to the 0.5 threshold, where a' is the attribute to change. For instance, take attributes a_1, a_2 and a_3. Let $p_{a_1}^Y = 0.49$, $p_{a_2}^Y = 0.1$ and $p_{a_3}^Y = 0.001$. a_1 will have a much larger probability of having value 1 than a_2 and a_3: the probability of the child node with the parent's a_1 being 1 will be larger than the one of the child with a_2 or a_3 set to 1. In a nutshell, we sort the attributes by their probability of changing from their most likely state, i.e., by $|p_a^Y - 0.5|$. Here, $|p_0^Y - 0.5| \leq |p_1^Y - 0.5| \leq |p_2^Y - 0.5|$.

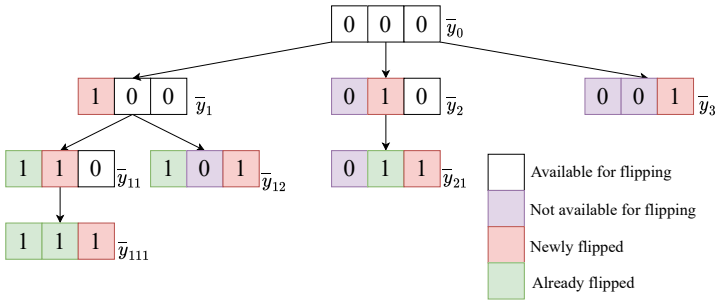


Fig. 5.3 An example of the realization tree. We assume that the changing probability decreases from the leftmost attribute to the rightmost one.

We repeat the procedure recursively on all child nodes, building the three with the depth-first search strategy. When we examine a node, we exclude those realizations already present on parents or siblings. In the example, consider the $\overline{y}_1$ node. In this case, the first attribute cannot be modified again, and $\overline{y}_1$ has only two children, $\overline{y}_{11}$ and $\overline{y}_{12}$. Similarly, when we land on $\overline{y}_2$, we cannot obtain a child by changing the first attribute, as this realization will have been already covered in the sibling $\overline{y}_1$.

We formalize the properties of the realization tree as follows:

- The probability of a parent realization is always greater than the probability of its children.
- The probabilities of siblings' realizations decrease from the most-probable-to-change to the least-probable-to-change.

These two properties allow us to adopt an efficient strategy to neglect unlikely realizations: given a node, its children and siblings on the right side have a lower or equal occurrence probability. Thus, we can efficiently prune the tree: if a node in the tree has a probability to be observed below a threshold, we prune all its children and rightmost siblings, returning to the parent and speeding up the computation.

To set this threshold, we start from the probability of a realization in a scenario where all the p_a^Y are equally probable – hence, all the realizations appear with the same probability: $1/2^{A_{eff}}$. We thus set the threshold to $\theta_2 \cdot 1/2^{A_{eff}}$, where θ_2 allows one to tune the trade-off between execution speed and model accuracy. Indeed, if the threshold is too high, not enough realizations will be considered, and the model will significantly differ from the true value. Conversely, with a low threshold, a multitude of potentially negligible realizations must be evaluated.

Notice that it is easy to recognize a poor choice of θ_2, by summing the probability of considered realizations (those belonging to the three after pruning), and imposing it to be at least – for instance – 0.98. In other words, we impose that:

$$\sum_{\overline{y}:p_{\overline{y}}\geq\frac{\theta_2}{2^{A_{eff}}}} p_{\overline{y}} \geq 0.98 \tag{5.7}$$

In our experiments, we use a greedy algorithm to find θ_2 such that Equation 5.7 holds.

5.2.3 Modeling Information loss

Applying z-anon to an input stream decreases the amount of information the output stream carries with respect to the original non-anonymized stream. There is a trade-off between data privacy and data usability: if no anonymization is in place, the information provided by the final dataset will be maximum. On the other hand, if the data are anonymized, privacy is protected but information is lost in the process.

Here we consider the entropy as a measure of the amount of information a dataset contains. This metric derives from Information Theory [125], which defines the information brought by the occurrence of a symbol among a set of possible symbols by knowing the probability of such symbol to appear.

In our case, each symbol is a possible realization of $\overline{Y}$. We can hence use Equation 5.4 to compute the amount of information of the release, by evaluating its entropy as:

$$I = -\sum_{\overline{y}} p_{\overline{y}} \log(p_{\overline{y}}). \tag{5.8}$$

By using Equation 5.8, it is also possible to evaluate the information loss caused by anonymizing a data stream, which can be computed as the difference between the information of the non-anonymized stream and the information of the anonymized one.

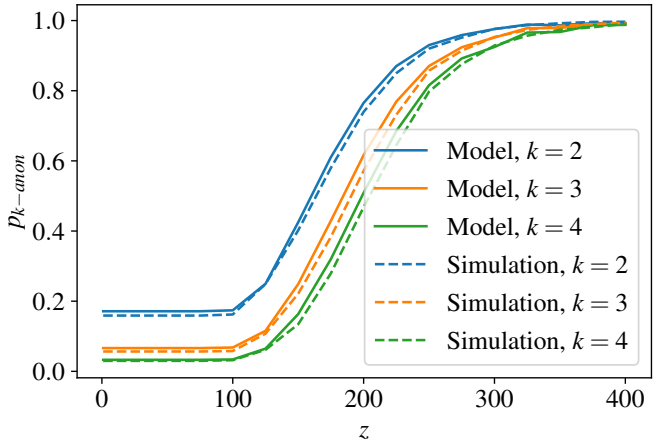


Fig. 5.4 p_{k-anon} changing z, for different k values. Exact model results and 10 iterations simulation averages are reported.

5.3 Mapping z-anonymity to k-anonymity

In the following, we use our model to show the impact of the system parameters on the z-anon and k-anon properties. Our model provides p_{k-anon} as a function of the scenario (U,A,λ) and system parameters ($z,k,\Delta t$, which are under our control). As such, this function provides the probability a generic user is k-anonymized in the released data. In the following analyses, where not otherwise noted, we use the parameters listed in Table 5.2.

5.3.1 The impact of z

We first focus on the impact of z. In Figure 5.4, we report how different values of z result in different probabilities for a given user to be k-anonymized – and compare the results to the simulation ones.

Different lines correspond to different values of k. The larger is z, the higher is p_{k-anon}. Focusing on $k=2$ (blue solid line), p_{k-anon} increases starting from $z=100$. With $z=250$, the probability of finding at least a user with an identical set of released attributes is already 0.8. When $z>350$, p_{k-anon} approaches 1, giving the almost certainty that the whole release is k-anonymized (with $k=2$). For $k=3,4$ (orange and blue line, respectively), p_{k-anon} exhibits a similar behaviour, with p_{k-anon} decreasing when increasing k, as it becomes harder to find k identical users

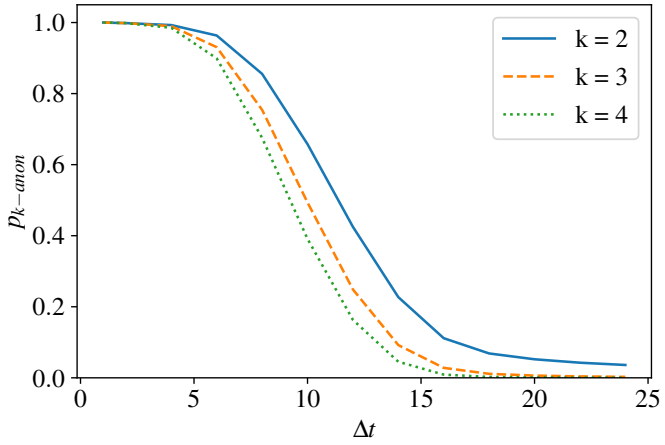


Fig. 5.5 p_{k-anon} changing Δt, for different k values.

by chance. In summary, one can tune z to enforce a desired k and p_{k-anon} on the released data.

Figure 5.4 also shows a comparison between the model and the result of simulations with the same parameters. To obtain the dashed lines (i.e., the simulation results), ten simulation are performed, with different seeds. For each of them, we evaluate p_{k-anon} for $k = 2,3,4$ and average the outcomes over all the simulations. The dashed lines and the corresponding solid ones follow the same trend and match almost perfectly - the differences being caused by the small number of simulation performed. This result also validates the essential correctness of the model: 5.3.5 provides more details on the simulation process and on its adherence to the model results.

5.3.2 The impact of Δt

The second design parameter one must set is Δt, the time window on which z-anon runs. In Figure 5.5, we show how p_{k-anon} varies while increasing Δt, with different values of k. Intuitively, the larger the time period, the lower the probability of a user being k-anonymized. A large time window allows also unpopular attributes to satisfy the z-anon property and be published, thus decreasing the p_{k-anon}. Conversely, with a narrow time window, only the most popular attributes result non z-private, with a positive effect on p_{k-anon}. In summary, Δt is another way for tuning the fraction of

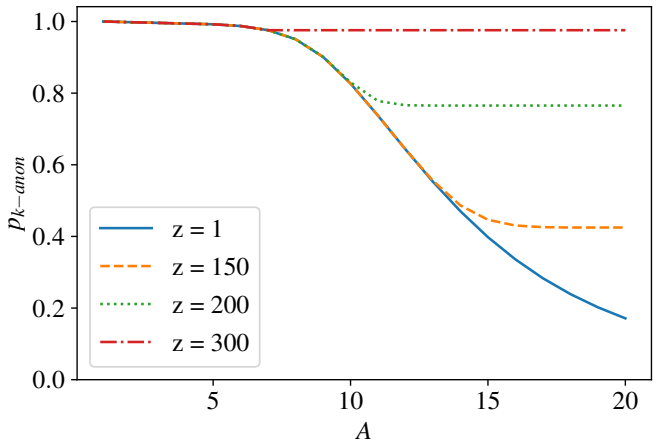
z-private attributes, with a direct impact on the k-anon of the released data. Since the effects of tuning z and Δt are interchangeable, from now on we will focus on z, knowing that acting on Δt would have an analogous effect. Note also that the choice of Δt impacts also the system memory and data structure size. As such, one would choose Δt on the specific use case, and regulate z for reaching the desired privacy level.

5.3.3 The impact of A

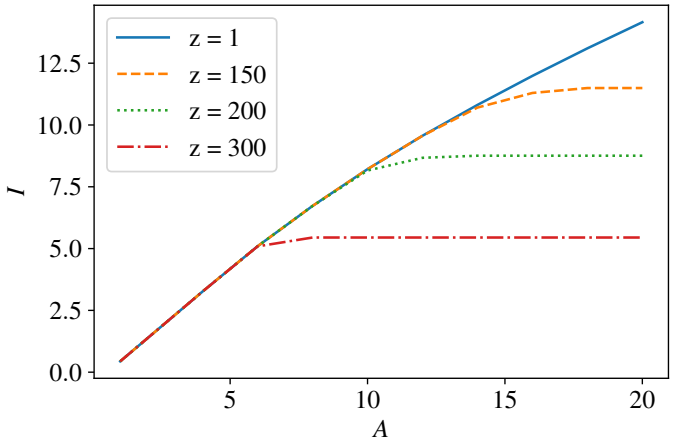
Then, we study the impact of the size of the catalog of attributes $A = |\mathscr{A}|$. In Figure 5.6a we show how the probability p_{k-anon} of a user being k-anonymized varies with A through the impact of z-anon. We consider a system where only the top A ranked attributes exist. As such, by increasing A, we add more and more infrequent attributes. Intuitively, a large number of attributes makes it hard to find users with the same output realization $\bar{y}$. However, with z-anon, the catalog size results limited and thus it plays a marginal role (see Figure 5.2), and, as such, infrequent attributes are rarely published.

In Figure 5.6a, p_{k-anon} starts at 1, when few attributes are present, and the number of their possible combinations is low. When A increases, less frequent attributes start to appear. The possible combinations of attributes explode exponentially. With $z = 1$, i.e., no z-anon in place, the probability of finding identical users rapidly goes to 0. Enabling z-anon, we prevent rare attributes from being released, thus limiting the number of combinations – see dashed lines.

Figure 5.6b shows the effect of the number of attributes on the entropy of the resulting dataset as defined in Equation 5.8, for different values of z. The entropy increases with the number of attributes A. If the attributes were equally probable (uniform distribution), the entropy will scale linearly with A. However, given the power law of attributes, the increase is sub-linear. Applying z-anon to the input data stream limits the growth of the entropy, preventing the appearance of infrequent combinations. Since least-likely attributes are protected by z-anon, they will not be published and will not add information to the final dataset. Therefore, the entropy tends to converge with higher A. The higher z, the fewer the released information. In other words, by tuning z (or Δt), it is possible to regulate the amount of information in the released data.



(a) p_{k-anon} changing A, considering different z values.



(b) The entropy I of the output z-anon dataset changing A, for different z values.

Fig. 5.6 The impact of A.

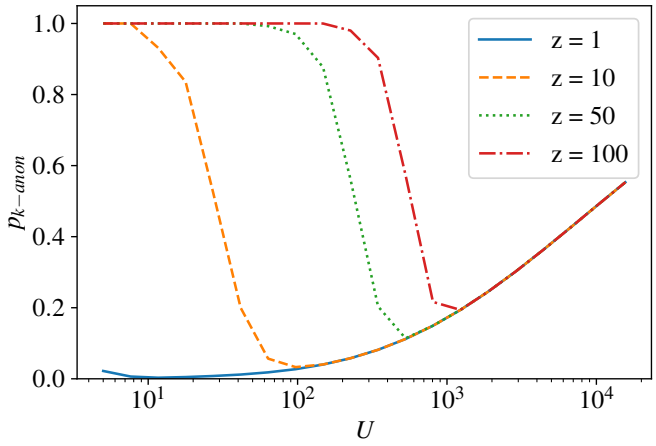
5.3.4 The impact of U

We now study how the number of users U impacts their probability to appear k-anonymized in the released data. In Figure 5.7a, we show how p_{k-anon} varies when increasing U, for different values of z. We notice the concurrence of two effects: first, with low values of U, even a small z ensures that users are k-anonymized, as rare realization have few chances to appear. Increasing the number of users leads to a decrease of p_{k-anon}. This happens because the large number of users causes even less-popular attributes to overcome the z threshold, hence increasing the number of possible combinations and, thus, decreasing p_{k-anon}. At a certain point (depending on z) all the attributes are likely to be published, and a second effect steps in: adding new users, each combination has a higher probability of appearing more than once, thus improving p_{k-anon}.

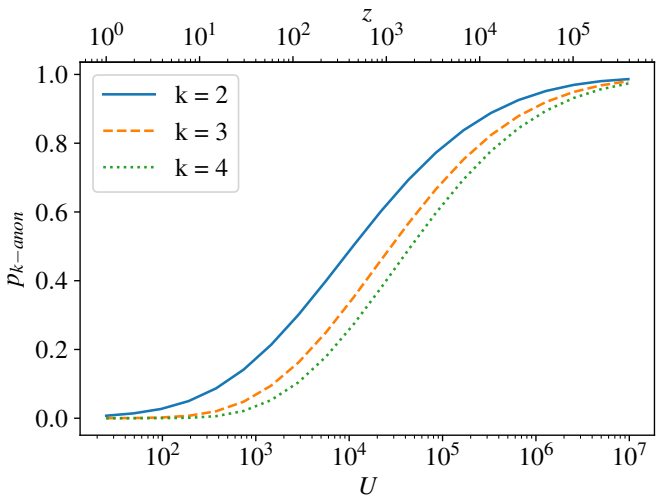
To prevent p_{k-anon} from decreasing with a large U, we now suppose we set z proportional with U. In Figure 5.7b, we show p_{k-anon} with increasing number of users (bottom x-axis) and, consequently, increasing z (top x-axis), as we set $z = U/25$. Focusing on the solid blue line ($k = 2$), we notice how p_{k-anon} grows with U, reaching values close to 1 with very large U (notice the log x-scale). With a higher k (dashed lines), the p_{k-anon} is only shifted to larger values of U. The figure shows that a large U leads to better guarantees of k-anon as far as z is set proportionally. As such, it is fundamental to consider the number of users in the system to properly set the z-anon parameters and, in turn, successfully achieve k-anon. Conversely, if z does not grow with U, performance guarantees worsen (see Figure 5.7a).

5.3.5 Model validation

To assess the validity of the model, we compare its results with those obtained by simulating the z-anon mechanism. To perform the simulation, we randomly generated an input trace that emulates a stream of tuples (t,u,a), with $U = 1\,000$, $A = 20$, $\lambda_{a_r} = {}^{0.2}/_{r}$. We then process the input trace via the z-anon mechanism, as described in Algorithm 1. At last, we collect the published tuples and evaluate the fraction of the 1000 users that result 2-anonymized as an estimate of p_{k-anon}.



(a) p_{k-anon} changing U, for different z values.



(b) p_{k-anon} increasing proportionally U and z ($z = U/25$), for different k values.

Fig. 5.7 The impact of U on p_{k-anon}.

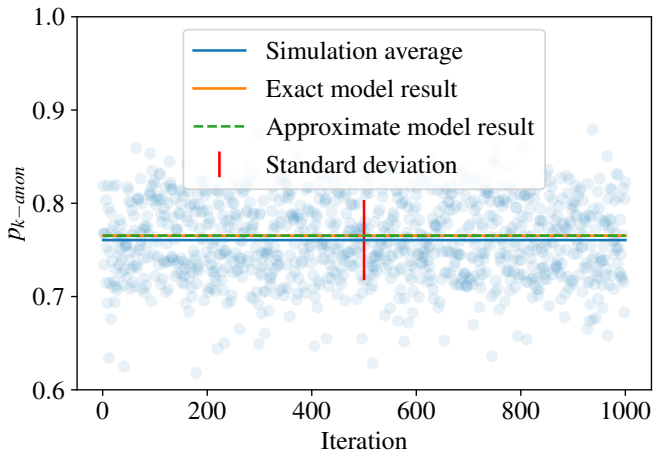


Fig. 5.8 The p_{k-anon} as evaluated by differently-seeded simulations, compared with the model results.

In Figure 5.8, we compare the results of the simulation with those of the model, considering both the exact and the approximated version. The complete list of scenario parameters is available in Table 5.2.

In Figure 5.8, each point represents the p_{k-anon} as obtained by each simulation, each with a different seed. The solid blue line indicates the average of the simulations while the solid orange and green lines report the estimation obtained by the exact and the approximated model, respectively. The last two return the same result. The average p_{k-anon} of the simulation results is within 0.005 from the exact model one, with a standard deviation of 0.04, indicated in Figure 5.8 as a vertical red bar.

It is worth noting that for simulation we take care of discarding the initial transient of duration Δt, during which the system starts accumulating observations, with no eviction happening.

5.4 Extension to user classes

The model we presented in Chapter 5.1.3 relies on the assumptions that the user activity rate is constant over time (i.e., it follows a homogeneous Poisson Process), their behaviour is independent (i.e., users' interactions do not depend one on the other), and homogeneous (i.e., every user acts in the same way). In this section, we relax the last assumption, introducing the concept of *classes* of users. We assume that

C classes exist and that each user belongs to one and one only class $c \in \{1,...,C\}$. Users in the same class behave homogeneously and potentially differently from those of other classes.

We consequently extend our model to consider the dependence on the class c of the user we are considering. In the notation, we will add the subscript of the class c to variables and probabilities. Hence, in each class c, there are U_c users.

The attribute exposing rate now depends on the user's class, thus $\lambda_{a,c}$. Consequently, $p^X_{a,c}$ is the probability a user in c exposes a in Δt. Let $p^O_{a,c}$ be the probability that this attribute a satisfies the z-anon constraint. This probability requires a different computation since it depends on the class c of the user and on users in other classes. The z-anon constraint is satisfied if there are at least $z-1$ other users, among all classes, that have exposed a in the past Δt. This can be written as in Equation 5.3 as the complementary event where the users exposing a do not add up to $z-1$.

To this end, we have to find all the possible combinations of users in the different classes exposing a. By denoting as $n_i \in \{0,\dots,U_i\}$ the number of users of class i that have exposed a given attribute a in the previous Δt, we can define the set $\mathscr{C}(z)$ of C-uples, whose sum does not exceed $z-2$:

$$\mathscr{C}(z) = \left\{ (n_1,\dots,n_C) : \sum_{i=1}^{C} n_i \leq z-2 \right\}$$

Then:

$$p^O_{a,c} = 1 - \sum_{(n_1,\dots,n_C)\in\mathscr{C}(z)} \left(\prod_{i=1}^{C} P\left[B_{i,c,a} = n_i\right] \right),$$

where $B_{i,c,a} \sim Binomial(U_i - \delta_{i,c}, p^X_{a,c})$ is the random variable representing the number of users in class i that have exposed a in the previous Δt. We remove one user when considering the same class c through the Kronecker delta $\delta_{i,c}$, as we already imposed that such user is exposing a.

Consequently, $p^Y_{a,c}$ is the probability a user in c publishes at least once a in Δt, and $p_{\bar{y},c}$ is the probability a user in c has the realization $\bar{y} = \{y_a\}_{a\in\mathscr{A}}$.

Finally, extending p_{k-anon} to $p_{k-anon,c}$ requires a few steps. Similar to what happens in $p^O_{a,c}$ the probability for a user of being k-anonymized depends on whether

it is possible to find in the release at least $k-1$ other users with the same realization – regardless of the class they belong to. We can reuse the definition of $\mathscr{C}$, now with parameter k, i.e., $\mathscr{C}(k)$. Then, the probability for a user in class c of being k-anonymized follows:

$$p_{k-anon,c} = \sum_{\bar{y}} \left\{ \left[1 - \sum_{(n_1,\ldots,n_C)\in\mathscr{C}(k)} \left(\prod_{i=1}^{C} P[Q_{i,c,\bar{y}} = n_i] \right) \right] \cdot p_{\bar{y},c} \right\},$$

where $Q_{i,c,\bar{y}} \sim Binomial(U_i - \delta_{i,c}, p_{\bar{y},i})$ is the random variable representing the number of users in class i with the same realization of attributes $\bar{y}$ in the previous Δt as our target user.

In the following, we explore two use cases considering two classes:

- Classes of activity: users belonging to one class are more active than users belonging to the other one;
- Classes of interest: users in different classes have different interests.

5.4.1 Classes of activity

In this scenario, users of the first class are more active than users of the second class. We define as $\lambda_{a,2}/\lambda_{a,1}$ the level of imbalance between classes. To provide a fair comparison, we want the overall average exposition rate λ_a to remain constant. This is verified if the following condition is satisfied:

$$U_1 \cdot \lambda_{a,1} + U_2 \cdot \lambda_{a,2} = (U_1 + U_2) \cdot \lambda_a,$$

where λ_a is the overall exposing rate of the users in $\mathscr{U}$ for a. Recall that U_1 and U_2 define respectively the number of users belonging to class 1 and the number of users belonging to class 2.

From the z-anon point of view, when $\lambda_{a,2} << \lambda_{a,1}$, this results in users of class 2 exposing few attributes, while the majority comes from users of class 1. Overall,

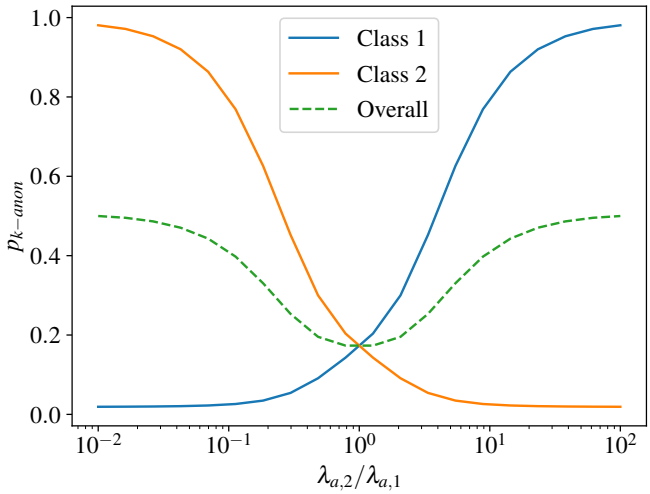


Fig. 5.9 p_{k-anon} with classes of activity ($z = 50$, $U_1 = 500$, $U_2 = 500$).

this implies that the "active" population is reduced, thus less tuples (t,u,a) can be published. This in turn will increase the probability of a user being k-anonymous. Figure 5.9 shows this effect showing $p_{k-anon,1}$, $p_{k-anon,2}$ and the resulting overall p_{k-anon}. The x-axis is log scale, and, as such, the Figure appears symmetric with respect to $\lambda_{a,2}/\lambda_{a,1} = 1$. Users are likely not to expose any attribute for the least active class, with thus a high probability of being k-anonymized. Conversely, p_{k-anon} decreases for the most active class, as its users are more active to compensate for the inactive users. Overall, the figure shows that p_{k-anon} benefits when the classes are strongly unbalanced (green dashed line). Conversely, the more similar the class rates become, the more the situation gets close to the single-class scenario. Indeed, when $\lambda_{a,2}/\lambda_{a,1} = 1$, the p_{k-anon} value reaches the same value shown in Figure 5.4.

5.4.2 Classes of interest

We consider a second use-case, where users belonging to one class are interested in a set of attributes, while the other users are interested in another set.

One possibility is to divide the attributes into two groups, for which the two classes of users have different interest, which we model with a different probability of publishing such attribute. We create two groups of attributes: recalling that λ_{a_r} is the exposing rate of attribute a in position r in the popularity rank, we assign

	$\lambda_{a_r,1}$	$\lambda_{a_r,2}$
Even attributes $\{a_r : r = 2k, k \in \mathbb{N}\}$	$\eta \cdot \lambda_{a_r}$	$(1-\eta) \cdot \lambda_{a_r}$
Odd attributes $\{a_r : r = 2k+1, k \in \mathbb{N}\}$	$(1-\eta) \cdot \lambda_{a_r}$	$\eta \cdot \lambda_{a_r}$

Table 5.3 Attributes rates for different classes of interest used in the example.

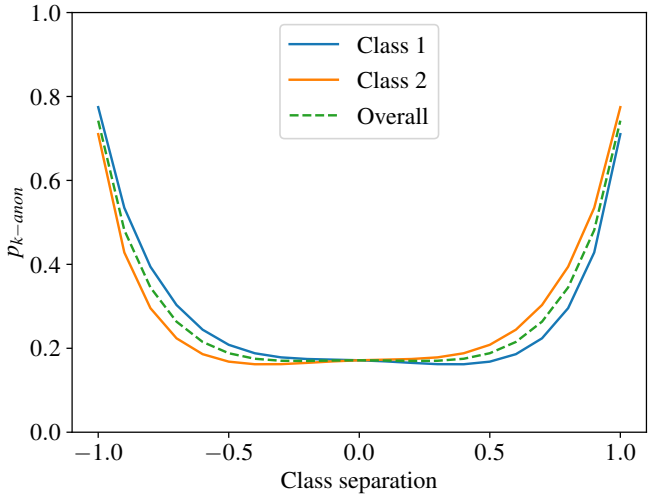


Fig. 5.10 p_{k-anon} with classes of interest ($z = 50$, $U_1 = 500$, $U_2 = 500$).

attributes in even positions of the rank to one group, and attributes in odd positions to the other group. Then, we assign the first group of attributes a rate $\eta \cdot \lambda_{a_r},\ \eta \in [0,1]$ to the first class of users; and $(1-\eta) \cdot \lambda_{a_r}$ to the second class of users. Conversely, we assign the second group of attributes a rate $(1-\eta) \cdot \lambda_{a_r}$ to the first class of users, and a rate $\eta \cdot \lambda_{a_r}$ to the second class of users. Table 5.3 formalizes the scenario of the example. If $\eta = 0.5$, the two classes become the same and, consequently, we obtain the same p_{k-anon} as for the single-class scenario. We define *class separation* as the difference $\eta - (1-\eta) = 2\eta - 1$.

In Figure 5.10, we show how p_{k-anon} varies with different *class separation* values. Similarly to the previous use case, splitting the users into classes increases the k-anon probability. In this case, though, both classes benefit from class separation. Increasing the class separation has the twofold effect of i) reducing the number of attributes that a user is likely to expose and ii) generating two dissimilar groups of users. As such, we conclude that a scenario where multiple groups of users that expose different attributes eases the achievement of k-anon.

As shown if Figure 5.6, the greater the p_{k-anon}, the lower the entropy of the released information. Although we do not present the figures for the sake of brevity, the case with classes of users follow the same principle. Where the classes are highly imbalanced in terms of either interest or activity (and the p_{k-anon} is larger) the quantity of information of the output decreases.

Chapter 6

Conclusion and Future Work

In this thesis, we presented several topics revolving around privacy-preserving management of data on the Web. We presented the role that Privacy Banners have in the current experience of the Web, and discussed the privacy guarantees of possible future solutions of Interest-Based Advertisement such as Topics API. Moreover, on the Privacy-Preserving Data Publishing hand, we have analyzed the privacy properties of the z-anonymity algorithm. In this final chapter, we discuss the finding of our work. In the following of the chapter we will draw conclusions on our work, and outline possible research paths for future works.

6.1 On Privacy Banners and beyond

Privacy Banners are ubiquitous in the current Web ecosystem, as their use has been forced by legislators in an effort of offering users control over the data that first and third parties collect about them on the Web. In Chapter 3.2, we showed that users, when facing a Privacy Banners, most users accept the use of cookies. Our results show that, in order to accurately measure the Web via crawling, one should take into account what happens to Web measures after accepting the Privacy Banner. However, the percentage of users accepting the use of cookies drastically decreases when users are offered a `Reject All` button. This leads to the conclusion that users which are offered a seamless solution to better protect their privacy are more likely to take action in that direction.

The results of Chapter 3.4 and Chapter 3.5 show that the metric obtained by crawling are very different whether the cookies are accepted or not. Upon accepting cookies, the crawler we designed detected a significantly larger number of trackers and third parties on the page, independently of the country or the category the website refers to. Moreover, pages become heavier and take more time to load. We believe these results highlight the need of careful crawling design in order to obtain results that are as close as possible to the true user experience on the Web, which in many cases includes the use of cookies.

However, it is also important to consider that cookies might be soon deprecated in favor of new framework that look for a better balance between data utility and users' privacy. Google's Topics API is an example of a new framework that could become a *de facto* standard for the industry, given the role of its proponent. In Chapter 4, we analyzed whether the privacy-preserving claims from Google are realistic, testing the Topics API against different re-identification attacks under a practical threat model. We conclude that, even if the Topics API are an improvement against the current third-party-cookies-based ecosystem, we still cannot rule out that attackers could reconstruct the identity of a user across multiple, colliding websites.

The results of this study pose in our opinion interesting question for the future in the field, such as whether the trade-off between data utility and users' privacy introduced by the Topics API can satisfy either the advertiser and the privacy advocates. One could also wonder how the utility performances of a contextual advertising framework — which makes no use of user data, and would resolve many of the privacy concerns — compare to a system based on Topics API. This, we believe, in as interesting line of research to explore.

6.2 Streaming data anonymization with z-anonymity

Finally, in Chapter 5 we discussed the privacy properties of an anonymization property and algorithm, the z-anonymity. To exactly measure the properties of z-anonymity, we mapped it with the k-anonymity. We tested the z-anon under many different parameter designs, such as the z itself, the number of users, the size of the attributes catalog, the division of the users in classes. As a side result, we introduced in the literature a statistical framework to estimate the probability that a dataset with binary attributes is k-anonymous, given the probabilities of the attributes being

either zero or one. For fair comparison, one should consider that the z-anon is a real-time, zero-delay anonymization techniques, and cannot benefit from the *a posteriori* knowledge of offline techniques such as k-anon.

This being said, the z-anon requires a high z threshold to become effective, which clearly affect the utility of the output data. Moreover, it is important to note that k-anon standards have been proved insufficient in different cases. In conclusion, that of an efficient, zero-delay anonymization technique for streaming data remains an open problem.

References

[1] Nick Nikiforakis, Alexandros Kapravelos, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *2013 IEEE Symposium on Security and Privacy*, pages 541–555. IEEE, 2013.

[2] Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. User tracking in the post-cookie era: How websites bypass gdpr consent to track users. In *Proceedings of the Web Conference 2021*, WWW '21, page 2130–2141, New York, NY, USA, 2021. Association for Computing Machinery.

[3] Aaron Cahn, Scott Alfeld, Paul Barford, and S. Muthukrishnan. An empirical study of web cookies. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 891–901, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee.

[4] Jonathan R. Mayer and John C. Mitchell. Third-party web tracking: Policy and technology. In *2012 IEEE Symposium on Security and Privacy*, pages 413–427. IEEE, 2012.

[5] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. Detecting and defending against Third-Party tracking on the web. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 155–168, San Jose, CA, April 2012. USENIX Association.

[6] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan Mayer, Arvind Narayanan, and Edward W. Felten. Cookies that give you away: The surveillance implications of web tracking. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 289–299, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee.

[7] Nayanamana Samarasinghe, Aashish Adhikari, Mohammad Mannan, and Amr Youssef. Et tu, brute? privacy analysis of government websites and mobile apps. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 564–575, New York, NY, USA, 2022. Association for Computing Machinery.

[8] Tomasz Bujlow, Valentín Carela-Español, Josep Sole-Pareta, and Pere Barlet-Ros. A survey on web tracking: Mechanisms, implications, and defenses. *Proceedings of the IEEE*, 105(8):1476–1510, 2017.

[9] Council of European Union. Directive 2009/136/EC amending Directive 2002/22/EC on universal service and users' rights relating to electronic communications networks and services, Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector and Regulation (EC) No 2006/2004 on cooperation between national authorities responsible for the enforcement of consumer protection laws. http://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:32009L0136 (Last accessed September 6, 2021), 2009.

[10] European Parliament and Council of European Union. Directive 95/46/EC. General Data Protection Regulation. http://data.consilium.europa.eu/doc/document/ST-5419-2016-INIT/en/pdf (Last accessed February 27, 2023), 2016.

[11] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. Privacy-Preserving Data Publishing: A Survey of Recent Developments. *ACM Comput. Surv.*, 42(4), June 2010.

[12] P. Samarati. Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[13] Nikhil Jha, Martino Trevisan, Marco Mellia, Rodrigo Irarrazaval, and Daniel Fernandez. I refuse if you let me: Studying user behavior with privacy banners at scale. In *2023 7th Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–9, 2023.

[14] Nikhil Jha, Martino Trevisan, Luca Vassio, and Marco Mellia. The internet with privacy policies: Measuring the web upon consent. *ACM Transactions on the Web (TWEB)*, 16(3):1–24, 2022.

[15] Nikhil Jha, Martino Trevisan, Emilio Leonardi, and Marco Mellia. On the robustness of topics api to a re-identification attack. In *Proceedings on Privacy Enhancing Technologies 2023(4)*, pages 66–78, 2023.

[16] Nikhil Jha, Thomas Favale, Luca Vassio, Martino Trevisan, and Marco Mellia. z-anonymity: Zero-Delay Anonymization for Data Streams. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 3996–4005, 2020.

[17] Nikhil Jha, Luca Vassio, Martino Trevisan, Emilio Leonardi, and Marco Mellia. Practical anonymization for data streams: z-anonymity and relation with k-anonymity. *Performance Evaluation*, 159:102329, 2023.

[18] Nikhil Jha, Martino Trevisan, Luca Vassio, Marco Mellia, Stefano Traverso, Alvaro Garcia-Recuero, Nikolaos Laoutaris, Amir Mehrjoo, Santiago Andrés Azcoitia, Ruben Cuevas Rumin, et al. A pims development kit for new personal data platforms. *IEEE Internet Computing*, 26(3):79–84, 2022.

[19] Hassan Metwalley, Stefano Traverso, Marco Mellia, Stanislav Miskovic, and Mario Baldi. The online tracking horde: a view from passive measurements. In *International Workshop on Traffic Monitoring and Analysis*, pages 111–125. Springer, 2015.

[20] Enric Pujol, Oliver Hohlfeld, and Anja Feldmann. Annoyed users: Ads and ad-block usage in the wild. In *Proceedings of the 2015 Internet Measurement Conference*, pages 93–106, 2015.

[21] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 674–689, 2014.

[22] Valentino Rizzo, Stefano Traverso, and Marco Mellia. Unveiling web fingerprinting in the wild via code mining and machine learning. *Proceedings on Privacy Enhancing Technologies*, 2021(1):43–63, 2021.

[23] Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. *User Tracking in the Post-Cookie Era: How Websites Bypass GDPR Consent to Track Users*, page 2130–2141. Association for Computing Machinery, New York, NY, USA, 2021.

[24] Martino Trevisan, Stefano Traverso, Eleonora Bassi, and Marco Mellia. 4 years of eu cookie law: Results and lessons learned. *Proc. Priv. Enhancing Technol.*, 2019(2):126–145, 2019.

[25] Rob van Eijk, Hadi Asghari, Philipp Winter, and Arvind Narayanan. The impact of user location on cookie notices (inside and outside of the european union). In *Workshop on Technology and Consumer Protection (ConPro'19)*, 2019.

[26] Célestin Matte, Nataliia Bielova, and Cristiana Santos. Do cookie banners respect my choice?: Measuring legal compliance of banners from iab europe's transparency and consent framework. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 791–809. IEEE, 2020.

[27] Iskander Sanchez-Rola, Matteo Dell'Amico, Platon Kotzias, Davide Balzarotti, Leyla Bilge, Pierre-Antoine Vervier, and Igor Santos. Can I Opt Out Yet? GDPR and the Global Illusion of Cookie Control. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, Asia CCS '19, page 340–351, New York, NY, USA, 2019. Association for Computing Machinery.

[28] California State Legislature. California Consumer Privacy Act of 2018. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375 (Last accessed May 25, 2021), 2018.

[29] Adrian Dabrowski, Georg Merzdovnik, Johanna Ullrich, Gerald Sendera, and Edgar Weippl. Measuring Cookies and Web Privacy in a Post-GDPR World. In David Choffnes and Marinho Barcellos, editors, *Passive and Active Measurement*, pages 258–270, Cham, 2019. Springer International Publishing.

[30] Thomas Linden, Rishabh Khandelwal, Hamza Harkous, and Kassem Fawaz. The privacy policy landscape after the gdpr. *Proceedings on Privacy Enhancing Technologies*, 2020(1):47–64, 2020.

[31] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. We value your privacy... now take some cookies. *Informatik Spektrum*, 42(5):345–346, 2019.

[32] Michael Kretschmer, Jan Pennekamp, and Klaus Wehrle. Cookie banners and privacy policies: Measuring the impact of the gdpr on the web. *ACM Trans. Web*, 15(4), jul 2021.

[33] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. We value your privacy... now take some cookies: Measuring the gdpr's impact on web privacy. In *26th Annual Network and Distributed System Security Symposium (NDSS '19)*. Internet Society, 2019.

[34] Maximilian Hils, Daniel W. Woods, and Rainer Böhme. Measuring the emergence of consent management on the web. In *Proceedings of the ACM Internet Measurement Conference*, IMC '20, page 317–332, New York, NY, USA, 2020. Association for Computing Machinery.

[35] Hana Habib, Sarah Pearman, Jiamin Wang, Yixin Zou, Alessandro Acquisti, Lorrie Faith Cranor, Norman Sadeh, and Florian Schaub. "it's a scavenger hunt": Usability of websites' opt-out and data deletion choices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery.

[36] Deloitte. Cookie Benchmark Study. https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/risk/deloitte-nl-risk-cookie-benchmark-study.pdf, 2020.

[37] Jan M Bauer, Regitze Bergstrøm, and Rune Foss-Madsen. Are you sure, you want a cookie?–the effects of choice architecture on users' decisions about sharing private online data. *Computers in Human Behavior*, 120:106729, 2021.

[38] Philip Hausner and Michael Gertz. Dark patterns in the interaction with cookie banners. *arXiv preprint arXiv:2103.14956*, 2021.

[39] Than Htut Soe, Oda Elise Nordberg, Frode Guribye, and Marija Slavkovik. Circumvention by design - dark patterns in cookie consent for online news outlets. In *Proceedings of the 11th Nordic Conference on Human-Computer*

Interaction: Shaping Experiences, Shaping Society, NordiCHI '20, New York, NY, USA, 2020. Association for Computing Machinery.

[40] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. Dark patterns after the gdpr: Scraping consent pop-ups and demonstrating their influence. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.

[41] Van Bavel R and Rodriguez Priego N. Testing the effect of the cookie banners on behaviour. (LF-NA-28287-EN-N), 2016.

[42] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. (un)informed consent: Studying gdpr consent notices in the field. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 973–990, New York, NY, USA, 2019. Association for Computing Machinery.

[43] Oksana Kulyk, Willard Rafnsson, Ida Marie Borberg, and Rene Hougard Pedersen. "so i sold my soul": Effects of dark patterns in cookie notices on end-user behavior and perceptions. In *Proceedings of 2022 Symposium on Usable Security and Privacy*. Internet society, 2022.

[44] Ashutosh Kumar Singh, Nisarg Upadhyaya, Arka Seth, Xuehui Hu, Nishanth Sastry, and Mainack Mondal. What cookie consent notices do users prefer: A study in the wild. In *Proceedings of the 2022 European Symposium on Usable Security*, pages 28–39, 2022.

[45] Tony Vila, Rachel Greenstadt, and David Molnar. Why we can't be bothered to read privacy policies models of privacy economics as a lemons market. In *Proceedings of the 5th international conference on Electronic commerce*, pages 403–407, 2003.

[46] Jens Grossklags and Nathan Good. Empirical studies on software notices to inform policy makers and usability designers. In *International Conference on Financial Cryptography and Data Security*, pages 341–355. Springer, 2007.

[47] Lynne M Coventry, Debora Jeske, John M Blythe, James Turland, and Pam Briggs. Personality and social framing in privacy decision-making: A study on cookie acceptance. *Frontiers in psychology*, 7:1341, 2016.

[48] Susanne Barth and Menno D.T. de Jong. The privacy paradox – investigating discrepancies between expressed privacy concerns and actual online behavior – a systematic literature review. *Telematics and Informatics*, 34(7):1038–1058, 2017.

[49] Xuehui Hu and Nishanth Sastry. Characterising third party cookie usage in the eu after gdpr. In *Proceedings of the 10th ACM Conference on Web Science*, pages 137–141, 2019.

[50] Jannick Sørensen and Sokol Kosta. Before and after gdpr: The changes in third party presence at public and private european websites. In *The World Wide Web Conference*, pages 1590–1600, 2019.

[51] HTTPArchive. https://httparchive.org (Last accessed September 6, 2021), 2021.

[52] Marjan Falahrastegar, Hamed Haddadi, Steve Uhlig, and Richard Mortier. The rise of panopticons: Examining region-specific third-party web tracking. In *International Workshop on Traffic Monitoring and Analysis*, pages 104–114. Springer, 2014.

[53] Steven Englehardt and Arvind Narayanan. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1388–1401, 2016.

[54] Costas Iordanou, Georgios Smaragdakis, Ingmar Poese, and Nikolaos Laoutaris. Tracing cross border web tracking. In *Proceedings of the Internet Measurement Conference 2018*, pages 329–342, 2018.

[55] Phani Vadrevu and Roberto Perdisci. What you see is not what you get: Discovering and tracking social engineering attack campaigns. In *Proceedings of the Internet Measurement Conference*, IMC '19, page 308–321, New York, NY, USA, 2019. Association for Computing Machinery.

[56] Waqar Aqeel, Balakrishnan Chandrasekaran, Anja Feldmann, and Bruce M. Maggs. 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*, IMC '20, page 680–695, New York, NY, USA, 2020. Association for Computing Machinery.

[57] Stefano Traverso, Martino Trevisan, Leonardo Giannantoni, Marco Mellia, and Hassan Metwalley. Benchmark and comparison of tracker-blockers: Should you trust them? In *2017 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–9. IEEE, 2017.

[58] Johan Mazel, Richard Garnier, and Kensuke Fukuda. A comparison of web privacy protection techniques. *Computer Communications*, 144:162–174, 2019.

[59] Satya Avasarala. *Selenium WebDriver practical guide*. Packt Publishing Ltd, 2014.

[60] Xiao Sophia Wang, Aruna Balasubramanian, Arvind Krishnamurthy, and David Wetherall. How speedy is SPDY? In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 387–399, Seattle, WA, April 2014. USENIX Association.

[61] Hugues de Saxcé, Iuniana Oprescu, and Yiping Chen. Is http/2 really faster than http/1.1? In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 293–299. IEEE, 2015.

[62] Enrico Bocchi, Luca De Cicco, and Dario Rossi. Measuring the quality of experience of web users. *ACM SIGCOMM Computer Communication Review*, 46(4):8–13, 2016.

[63] Jeffrey Erman, Vijay Gopalakrishnan, Rittwik Jana, and Kadangode K Ramakrishnan. Towards a spdy'ier mobile web? *IEEE/ACM Transactions on Networking*, 23(6):2010–2023, 2015.

[64] Özgü Alay, Andra Lutu, Miguel Peón-Quirós, Vincenzo Mancuso, Thomas Hirsch, Kristian Evensen, Audun Hansen, Stefan Alfredsson, Jonas Karlsson, Anna Brunstrom, et al. Experience: An open platform for experimentation with commercial mobile broadband networks. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 70–78, 2017.

[65] Alemnew Sheferaw Asrese, Ermias Andargie Walelgne, Vaibhav Bajpai, Andra Lutu, Özgü Alay, and Jörg Ott. Measuring web quality of experience in cellular networks. In *International Conference on Passive and Active Network Measurement*, pages 18–33. Springer, 2019.

[66] Ashiwan Sivakumar, Shankaranarayanan Puzhavakath Narayanan, Vijay Gopalakrishnan, Seungjoon Lee, Sanjay Rao, and Subhabrata Sen. Parcel: Proxy assisted browsing in cellular networks for energy and latency reduction. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pages 325–336, 2014.

[67] Xiao Sophia Wang, Arvind Krishnamurthy, and David Wetherall. Speeding up web page loads with shandian. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 109–122, Santa Clara, CA, March 2016. USENIX Association.

[68] Vaspol Ruamviboonsuk, Ravi Netravali, Muhammed Uluyol, and Harsha V Madhyastha. Vroom: Accelerating the mobile web with server-aided dependency resolution. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 390–403, 2017.

[69] Ravi Netravali, Anirudh Sivaraman, Somak Das, Ameesh Goyal, Keith Winstein, James Mickens, and Hari Balakrishnan. Mahimahi: Accurate record-and-replay for HTTP. In *2015 USENIX Annual Technical Conference (USENIX ATC 15)*, pages 417–429, Santa Clara, CA, July 2015. USENIX Association.

[70] Pelayo Vallina, Álvaro Feal, Julien Gamba, Narseo Vallina-Rodriguez, and Antonio Fernández Anta. Tales from the porn: A comprehensive privacy analysis of the web porn ecosystem. In *Proceedings of the Internet Measurement*

Conference, IMC '19, page 245–258, New York, NY, USA, 2019. Association for Computing Machinery.

[71] Jordan Jueckstock, Shaown Sarker, Peter Snyder, Aidan Beggs, Panagiotis Papadopoulos, Matteo Varvello, Benjamin Livshits, and Alexandros Kapravelos. *Towards Realistic and ReproducibleWeb Crawl Measurements*, page 80–91. Association for Computing Machinery, New York, NY, USA, 2021.

[72] I don't care about cookies. https://www.i-dont-care-about-cookies.eu/ (Last accessed September 6, 2021), 2021.

[73] Remove Cookie Banners. https://chrome.google.com/webstore/detail/remove-cookie-banners/pacehjmodmfilemfbcahnpdcdmlocjnm (Last accessed September 6, 2021), 2021.

[74] Ninja Cookie. https://ninja-cookie.com/ (Last accessed September 6, 2021), 2021.

[75] Cliqz AutoConsent. https://github.com/cliqz-oss/autoconsent (Last accessed September 6, 2021), 2021.

[76] Consent-O-Matic. https://github.com/cavi-au/Consent-O-Matic (Last accessed September 6, 2021), 2021.

[77] Stephen Farrell and Hannes Tschofenig. Pervasive Monitoring Is an Attack. RFC 7258, May 2014.

[78] Janice C Sipior, Burke T Ward, and Ruben A Mendoza. Online privacy concerns associated with cookies, flash cookies, and web beacons. *Journal of internet commerce*, 10(1):1–16, 2011.

[79] José Estrada-Jiménez, Javier Parra-Arnau, Ana Rodríguez-Hoyos, and Jordi Forné. Online advertising: Analysis of privacy threats and protection approaches. *Computer Communications*, 100:32–51, 2017.

[80] Deepak Ravichandran and S Vasilvitskii. Evaluation of cohort algorithms for the floc api. *Google Research & Ads white paper*, 2021.

[81] Alessandro Epasto, Andrés Muñoz Medina, Steven Avery, Yijian Bai, Robert Busa-Fekete, CJ Carey, Ya Gao, David Guthrie, Subham Ghosh, James Ioannidis, Junyi Jiao, Jakub Lacki, Jason Lee, Arne Mauser, Brian Milch, Vahab Mirrokni, Deepak Ravichandran, Wei Shi, Max Spero, Yunting Sun, Umar Syed, Sergei Vassilvitskii, and Shuo Wang. Clustering for private interest-based advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 2802–2810, New York, NY, USA, 2021. Association for Computing Machinery.

[82] Eric Rescorla and Martin Thomson. Technical comments on floc privacy. 2021.

[83] Alex Berke and Dan Calacci. Privacy limitations of interest-based advertising on the web: A post-mortem empirical analysis of google's floc. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 337–349, New York, NY, USA, 2022. Association for Computing Machinery.

[84] Florian Turati. Analysing and exploiting google's floc advertising proposal. Master's thesis, ETH Zurich, Department of Computer Science, 2022.

[85] Lukasz Olejnik, Claude Castelluccia, and Artur Janc. Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns. In *5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*, Spain, July 2012.

[86] Dominik Herrmann, Christian Banse, and Hannes Federrath. Behavior-based tracking: Exploiting characteristic patterns in dns traffic. *Computers & Security*, 39:17–33, 2013.

[87] Luca Vassio, Danilo Giordano, Martino Trevisan, Marco Mellia, and Ana Paula Couto da Silva. Users' Fingerprinting Techniques from TCP Traffic. In *Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, Big-DAMA '17, page 49–54, New York, NY, USA, 2017. Association for Computing Machinery.

[88] Audrey Randall, Peter Snyder, Alisha Ukani, Alex C. Snoeren, Geoffrey M. Voelker, Stefan Savage, and Aaron Schulman. Measuring uid smuggling in the wild. In *Proceedings of the 22nd ACM Internet Measurement Conference*, page 230–243, New York, NY, USA, 2022. Association for Computing Machinery.

[89] Alessandro Epasto, Andres Munoz Medina, Christina Ilvento, and Josh Karlin. Measures of cross-site re-identification risk: An analysis of the topics api proposal. https://github.com/patcg-individual-drafts/topics/blob/main/topics_analysis.pdf, 2022.

[90] Martin Thomson. A privacy analysis of google's topics proposal. 2023.

[91] CJ Carey, Travis Dick, Alessandro Epasto, Adel Javanmard, Josh Karlin, Shankar Kumar, Andres Muñoz Medina, Vahab Mirrokni, Gabriel Henrique Nunes, Sergei Vassilvitskii, and Peilin Zhong. Measuring re-identification risk. *Proc. ACM Manag. Data*, 1(2), jun 2023.

[92] Latanya Sweeney. Guaranteeing anonymity when sharing medical data, the Datafly System. In *Proceedings of the AMIA Annual Fall Symposium*, page 51. American Medical Informatics Association, 1997.

[93] A. Narayanan and V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125, 2008.

[94] J. Cao, B. Carminati, E. Ferrari, and K. Tan. CASTLE: Continuously Anonymizing Data Streams. *IEEE Transactions on Dependable and Secure Computing*, 8(3):337–352, 2011.

[95] J. Li, B. C. Ooi, and W. Wang. Anonymizing Streaming Data for Privacy Protection. In *2008 IEEE 24th International Conference on Data Engineering*, pages 1367–1369, 2008.

[96] Bin Zhou, Yi Han, Jian Pei, Bin Jiang, Yufei Tao, and Yan Jia. Continuous Privacy Preserving Publishing of Data Streams. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, page 648–659, New York, NY, USA, 2009. ACM.

[97] J. Zhang, J. Yang, J. Zhang, and Y. Yuan. KIDS:K-anonymization data stream base on sliding window. In *2010 2nd International Conference on Future Computer and Communication*, volume 2, pages 311–316, 2010.

[98] Jimmy Tekli, Bechara Al Bouna, Youssef Bou Issa, Marc Kamradt, and Ramzi Haraty. (k, l)-Clustering for Transactional Data Streams Anonymization. In *International Conference on Information Security Practice and Experience*, pages 544–556. Springer, 2018.

[99] A. B. Sakpere and A. V. D. M. Kayem. Adaptive buffer resizing for efficient anonymization of streaming data with minimal information loss. In *2015 International Conference on Information Systems Security and Privacy (ICISSP)*, pages 1–11, 2015.

[100] Ankhbayar Otgonbayar, Zeeshan Pervez, Keshav Dahal, and Steve Eager. K-VARP: K-anonymity for varied data streams via partitioning. *Information Sciences*, 467:238–255, October 2018.

[101] A. Otgonbayar, Z. Pervez, and K. Dahal. Toward Anonymizing IoT Data Streams via Partitioning. In *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 331–336, 2016.

[102] M. Khavkin and M. Last. Preserving Differential Privacy and Utility of Non-stationary Data Streams. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 29–34, 2018.

[103] J. Domingo-Ferrer, J. Soria-Comas, and R. Mulero-Vellido. Steered Microaggregation as a Unified Primitive to Anonymize Data Sets and Data Streams. *IEEE Transactions on Information Forensics and Security*, 14(12):3298–3311, 2019.

[104] M.A.P. Chamikara, P. Bertok, D. Liu, S. Camtepe, and I. Khalil. An efficient and scalable privacy preserving algorithm for big data and data streams. *Computers & Security*, 87:101570, 2019.

[105] Soohyung Kim, Min Kyoung Sung, and Yon Dohn Chung. A framework to preserve the privacy of electronic health data streams. *Journal of Biomedical Informatics*, 50:95 – 106, 2014. Special Issue on Informatics Methods in Medical Privacy.

[106] Saad A. Abdelhameed, Sherin M. Moussa, and Mohamed E. Khalifa. Restricted Sensitive Attributes-based Sequential Anonymization (RSA-SA) approach for privacy-preserving data stream publishing. *Knowledge-Based Systems*, 164:1 – 20, 2019.

[107] J. Wang, C. Deng, and X. Li. Two Privacy-Preserving Approaches for Publishing Transactional Data Streams. *IEEE Access*, 6:23648–23658, 2018.

[108] Diego Neves da Hora, Alemnew Sheferaw Asrese, Vassilis Christophides, Renata Teixeira, and Dario Rossi. Narrowing the gap between qos metrics and web qoe using above-the-fold metrics. In *International Conference on Passive and Active Network Measurement*, pages 31–43. Springer, 2018.

[109] SpeedIndex. https://web.dev/speed-index/ (Last accessed September 6, 2021), 2021.

[110] Docker. https://www.docker.com/ (Last accessed September 6, 2021), 2021.

[111] SimilarWeb. https://www.similarweb.com (Last accessed September 6, 2021), 2021.

[112] Antoine Vastel, Walter Rudametkin, Romain Rouvoy, and Xavier Blanc. Fp-crawlers: studying the resilience of browser fingerprinting to block crawlers. In *MADWeb'20-NDSS Workshop on Measurements, Attacks, and Defenses for the Web*, 2020.

[113] SimilarWeb. https://www.similarweb.com/category (Last accessed January 31, 2022), 2022.

[114] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. *arXiv preprint arXiv:1806.01156*, 2018.

[115] WhoTracks.me. https://whotracks.me/ (Last accessed September 6, 2021), 2021.

[116] EasyPrivacy. https://easylist.to/easylist/easyprivacy.txt (Last accessed September 6, 2021), 2021.

[117] AdGuard. https://adguard.com/ (Last accessed September 6, 2021), 2021.

[118] Hassan Metwalley, Stefano Traverso, and Marco Mellia. Using passive measurements to demystify online trackers. *Computer*, 49(3):50–55, 2016.

[119] Nikhil Jha, Martino Trevisan, Emilio Leonardi, and Marco Mellia. Re-identification attacks against the Topics API. *Submitted to ACM Transactions on Privacy and Security*.

[120] Luc Devroye. Sample-based non-uniform random variate generation. In *Proceedings of the 18th conference on Winter simulation*, pages 260–265, 1986.

[121] Bimal Viswanath, M Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. Towards detecting anomalous user behavior in online social networks. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 223–238, 2014.

[122] T. Favale, M. Trevisan, I. Drago, and M. Mellia. α-mon: Traffic anonymizer for passive monitoring. *IEEE Transactions on Network and Service Management*, pages 1–1, 2021.

[123] Adam Meyerson and Ryan Williams. On the Complexity of Optimal K-Anonymity. In *Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '04, page 223–228, New York, NY, USA, 2004. Association for Computing Machinery.

[124] Lada A. Adamic and Bernardo A. Huberman. Power-law distribution of the world wide web. *Science*, 287(5461):2115–2115, 2000.

[125] Leon Brillouin. *Science and information theory*. Courier Corporation, 2013.