POLITECNICO DI TORINO
Repository ISTITUZIONALE

Re-Identification Attacks against the Topics API

Original
Re-Identification Attacks against the Topics API / Jha, Nikhil; Trevisan, Martino; Leonardi, Emilio; Mellia, Marco. - In: ACM TRANSACTIONS ON THE WEB. - ISSN 1559-1131. - ELETTRONICO. - 18:3(2024), pp. 1-24. [10.1145/3675400]

Availability:
This version is available at: 11583/2994443 since: 2024-11-15T13:23:31Z

Publisher:
ASSOC COMPUTING MACHINERY

Published
DOI:10.1145/3675400

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

30 May 2026

Re-Identification Attacks against the Topics API

NIKHIL JHA, Politecnico di Torino, Torino, Italy
MARTINO TREVISAN, Università degli Studi di Trieste, Trieste, Italy
EMILIO LEONARDI, Politecnico di Torino, Torino, Italy
MARCO MELLIA, Politecnico di Torino, Torino, Italy

Recently, Google proposed the Topics API framework as a privacy-friendly alternative for behavioural advertising as a possible solution to balance user's privacy and advertisement effectiveness. Using the Topics API, the browser builds a user profile based on navigation history, which advertisers can access. The Topics API aim at becoming the new standard for behavioural advertising, thus it is necessary to fully understand its operation and find possible limitations. In this article, we evaluate the robustness of the Topics API to a re-identification attack. To build a user profile, we suppose an attacker accumulates over time the topics a user exposes to different websites. The attacker later re-identifies the same user matching the profiles of their audience. We leverage real traffic traces and realistic population models, and we present increasingly powerful attack threats. We find that the Topics API mitigates but cannot prevent re-identification from taking place, as there is a sizeable chance that a user's profile remains unique within a website's audience and the attacker successfully matches it with the profile of the same user on a second website. Depending on environmental factors, the probability of correct re-identification can reach 50%, considering a pool of 1, 000 users. We offer the code and data we use in this work to stimulate further studies and the tuning of the Topic API parameters.[1]

CCS Concepts: • **Security and privacy → Privacy-preserving protocols;**

Additional Key Words and Phrases: Web privacy, anonymity, behavioral advertising, topics API

ACM Reference Format:
Nikhil Jha, Martino Trevisan, Emilio Leonardi, and Marco Mellia. 2024. Re-Identification Attacks against the Topics API. *ACM Trans. Web* 18, 3, Article 39 (August 2024), 24 pages. https://doi.org/10.1145/3675400

1 Introduction

In the current Web ecosystem, targeted or behavioural advertising lets providers monetize their content, by collecting and processing personal data to build accurate user profiles. Advertisers rely

[1]A previous version of this work appeared at the 2023 Privacy Enhancing Technologies Symposium [13].

This work was partially supported by the SERICS project (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU and the project "National Center for HPC, Big Data and Quantum Computing", CN00000013 (Bando M42C –Investimento 1.4 – Avviso Centri Nazionali" – D.D. n. 3138 of 16.12.2021, funded with MUR Decree n. 1031 of 17.06.2022).
Authors' Contact Information: Nikhil Jha, Politecnico di Torino, Torino, Piemonte, Italy; e-mail: nikhil.jha@polito.it; Martino Trevisan, Università degli Studi di Trieste, Trieste, Friuli-Venezia Giulia, Italy; e-mail: martino.trevisan@dia.units.it; Emilio Leonardi, Politecnico di Torino, Torino, Piemonte, Italy; e-mail: emilio.leonardi@polito.it; Marco Mellia, Politecnico di Torino, Torino, Piemonte, Italy; e-mail: marco.mellia@polito.it.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 1559-1131/2024/08-ART39
https://doi.org/10.1145/3675400

on the construction of profiles based on the websites a user visits [6, 15, 16]. As well known, third-party cookies allow tracking platforms to identify the same user on different websites: when a user visits a website, a tracker installs a third-party *profiling* cookie on the user's client. The cookie acts as a unique identifier that lets the tracker identify the user on subsequent visits to any websites where the third-party tracker is present.As such, the tracker learns the sequence of websites the given user visits and builds a profile describing their interests. Such profiles are used to provide personalized advertisements. In some cases, tracking platforms employ more sophisticated and privacy-intrusive techniques such as browser fingerprinting or ID synchronization [18, 23].

This massive data collection has created tension between users and the ads ecosystem [9, 15, 24]. Mozilla Firefox and Apple Safari have started battling third-party cookies by giving third-party cookies a separate cookie jar per site, so they cannot be used to track users across sites anymore. In parallel, leading researchers and industries are studying new paradigms that are more respectful of users' privacy while enabling targeted advertising. Novel proposals have one common feature: the introduction of new techniques that limit or let the user control the amount of disclosed personal information. Google proposed the **Federated Learning of Cohorts** (**FLoC**) [21]. FLoC clusters users in cohorts according to their interests, computed by each one's browser based on the user's recent activity. In the proponents' intentions, this solution should have prevented tracking, as every user was "hidden" in their cohort. The main issue is that while a user could hide inside a cohort for a short period of time, the sequence of cohorts they belonged to across time could create an identifier, increasingly unique [22]. Eventually, Google replaced FLoC with a new proposal called *Topics API*. With the Topics API, the browser is in charge of building the set of topics the user is interested in based on their navigation history. Websites ask for such topics to serve targeted advertisements or services. Namely, when a website asks, the Topic API returns up to three topics per week, possibly replacing one real topic with a random one.

The Topics API framework is meant to become the new standard for behavioural advertising. At the moment of writing, Google started deprecating the use of cookies for 1% of Chrome users in the first quarter of 2024, and plans to reach 100% in the third quarter of the same year.[2] It is thus urgent to fully understand the operation of the Topics API, and independent researchers must verify the strengths and weaknesses of such an approach as done by Mozilla [25] and Google [4, 7].

In this article, we provide an independent evaluation of the Topics API. We build and extend our previous work [13] by providing a more comprehensive set of attacks and means to measure their effectiveness. Using a data-driven approach, we build realistic population models that we use to quantify the feasibility of a re-identification attack. We assume that an attacker (i) builds the victims' profile by accumulating the topics exposed to a given website over weeks, and then (ii) tries to re-identify the same victim among the audience of a second website by comparing the profiles—as studied by authors of [4, 7]. If successful, such an attack would allow platforms to track users across websites. Generalising the attack sketched by Thomson [25], we use a threshold mechanism which effectively excludes random topics and filters out rare topics that impair re-identification.

We contribute to three main results:

(1) We show that the introduction of the Topics API algorithm cannot prevent re-identification. Depending on the website's audience size and heterogeneity, up to 50% of users still let the attacker reconstruct a profile that allows re-identification when matched on a second population.
(2) We demonstrate that the replacement of actual topics with random ones has limited effect in preventing the attack. Yet, a simple denoising algorithm is very efficient in removing random topics from the profiles the attacker builds.

[2] https://developer.chrome.com/blog/cookie-countdown-2023oct/, accessed on August 16, 2024

Table 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
$\mathcal{B}_{u,e}$	Bag of visited *websites* by user u at epoch e
$\mathcal{T}_{u,e}$	Bag of visited *topics* by user u at epoch e
$\mathcal{P}_{u,e}$	Profile for the user u at epoch e
$\mathcal{P}_{u,e,w}$	Exposed Profile to website w for user u at epoch e
$\mathcal{G}_{u,N,w}$	Global Reconstructed Profile by w after N epochs
$\mathcal{R}^{f}_{u,N,w}$	Denoised Reconstructed Profile by w after N epochs with threshold f

(3) We compare different attacks to re-identify a user. The re-identification attack that we devise can top 50% with limited **false positives** (**FP**), less than 8%; or can decrease to 25% but with practically no FP. However, it is also important to consider that such probabilities are a function of the attacker's observation period and that many weeks may be needed to carry out the attack in practice.

Our study highlights the need for continued research and development of privacy-preserving advertising techniques to ensure that user privacy is respected in the digital age. To foster research in this field, we release the code and data to replicate and extend our experiments.[3]

In our previous work [13], we studied the probability of a user exposing a unique combination of topics, focusing on k-anonymity. In this article, we focus directly on the re-identification probability by mounting new and more general attacks and extensively study the impact of environmental and design parameters. In addition, we improve the denoising algorithm of [13] and study in depth the effect that different filtering thresholds have on the performances of the attack.

The remainder of the article is organized as follows: Section 2 formalizes Topics API operation. Section 3 details the threat model and the attacks evaluated in this work. In Sections 4 and 5, we describe the dataset and models we use to generate synthetic populations, respectively. Section 6 shows the results in terms of attack effectiveness, while Section 7 studies the impact of Topics API design parameters. Finally, Section 8 summarizes related work, and Section 9 discusses our findings, future directions and possible improvements to the Topics API.

2 The Topics API

In this section, we describe how the Topics API operates for creating a profile from the user's browsing history. We sum up the relevant terminology in Table 1. We consider a browser that a user employs to navigate the Internet.[4] 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* $\mathcal{B}_{u,e}$ for the user u. It keeps track only of the website hostnames the user *intentionally* visited, for example, 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

[3]The code is available at https://github.com/nikhiljha95/topics-api-simulator.
[4]We intentionally confuse the terms *user* and *browser* to identify the person and the application they use to navigate the Internet.

e, let $\mathcal{B}_{u,e} = \{(w_1, f_{1,u,e}), (w_2, f_{2,u,e}), \ldots, (w_n, f_{n,u,e})\}$, 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 $\mathcal{B}_{u,e}$ over the past E epochs to create a corresponding *Exposed Profile* $\mathcal{P}_{u,e,w}$ 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.[5] The operation of the Topics API has the following steps.

Step 1 - From websites to topics. For each of the websites $w_i \in \mathcal{B}_{u,e}$, 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 $\mathcal{B}_{u,e}$ is transformed into a *topic history* $\mathcal{T}_{u,e} = \{(t_1, f'_{1,u,e}), (t_2, f'_{2,u,e}), \ldots, (t_m, f'_{m,u,e})\}$, 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 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).[6] In our experiments, we employ the Google ML model implemented in Chrome. In its first implementation, it supports $n_{topic} = 349$ topics[7] and the model is based on a Neural Network trained by Google using a manually curated set of 10,000 domains.[8] It leverages website hostnames only and neglects any other part of a URL.[9]

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

Step 3 - Per-website topic selection. The first time the user visits the website w, the browser generates a *Exposed Profile* $\mathcal{P}_{u,e,w}$. For each past epoch $i \in \{e-1, \ldots, e-E\}$, the browser selects at random one topic t_i^* from the profile history $\mathcal{P}_{u,i}$. To increase privacy guarantees, with probability p the browser replaces the topic t_i^* with a random topic t_{rnhd} uniformly selected from the global topic list. p is currently suggested to be 0.05. $\mathcal{P}_{u,e,w}$ contains thus at most E topics (a topic picked from $\mathcal{P}_{u,e-1}$, a topic from $\mathcal{P}_{u,e-2}$, 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 $\mathcal{P}_{u,e,w}$ and use the returned topics to provide behavioural advertising. Notice that the Exposed Profile $\mathcal{P}_{u,e,w}$ is built only for websites intentionally (first-party) visited by the user u. Any third-party service (e.g., a component embedded on the webpage of site w, but hosted on a different domain)

[5] https://developer.chrome.com/docs/privacy-sandbox/topics/, accessed on August 16, 2024
[6] https://iabtechlab.com/standards/audience-taxonomy/, accessed on August 16, 2024
[7] https://github.com/patcg-individual-drafts/topics/blob/main/taxonomy_v1.md, accessed on August 16, 2024
[8] 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.
[9] 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 August 16, 2024

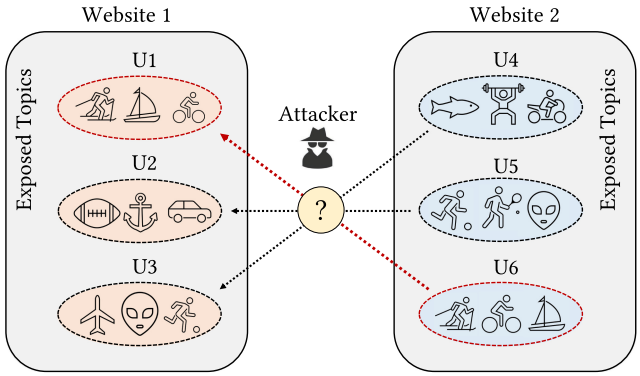


Fig. 1. Threat model sketch: An attacker leverages the Exposed Profiles obtained from the Topics API to re-identify the same user in the population of two websites.

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 $\mathcal{P}_{u,e,w}$ of w.

Periodic Profile update. At the beginning of the epoch $e+1$, the browser computes the new profile history $\mathcal{P}_{u,e+1}$ and discards $\mathcal{P}_{u,e-E}$. Similarly, if and when the user visits again the website w, the browser creates $\mathcal{P}_{u,e+1,w}$ from $\mathcal{P}_{u,e,w}$: it includes a new topic selected from $\mathcal{P}_{u,e+1}$ (Step 3), and removes the oldest topic, that is, the one originally belonging to $\mathcal{P}_{u,e-E+1}$ (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).

3 Attacks Against the Topics API

3.1 Threat Model

In this article, we consider the threat model introduced by the same proponents of Topics API [4, 7] and discussed in a technical report by Mozilla [25]. In detail, we consider the risk of re-identification—that is, 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 [12, 13, 17, 27]. We sketch the threat model in Figure 1.

As in [4, 7], 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 $\mathcal{P}_{u,e,w}$ 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* $\mathcal{G}_{u,N,w}$, where $\mathcal{G}_{u,N,w} = \cup_{e\in[1,N]}\mathcal{P}_{u,e,w}$.[10] 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.

[10]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.

Notice that this attack may be carried out by a third-party service s. Let us assume that both w_1 and w_2 embed s. By the means of the partitioned third-party cookies — which will still be usable, even in the new ad paradigm—the third party is thus able to collect first-party-wise topics, without being able to link them cross-site. The third party then builds $\mathcal{G}_{u,N,w_1}$ and $\mathcal{G}_{u,N,w_2}$ autonomously so that it can match the profiles of users in both audiences.[11]

In a real-world scenario, we expect the threat model to be more effective if two websites have a large portion of co-occurring users. This could happen if both websites are very popular or if both websites belong to a niche context or offer local services (e.g., for shops, restaurants, etc.). In the case of popular websites, techniques to reduce the population of possible matching users could be used to partition the audience to increase the re-identification probability.

3.2 Random and Rare Topic Denoising

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

Preliminary studies on the Topics API, such as [25], mainly considered the need for identifying the random topics in an Exposed Profile $\mathcal{P}_{u,e,w}$ by carrying out a simple statistical test based on the number of times a topic is exposed by a user. The author of [25] states that observing a topic more than once is sufficient to infer its authenticity with high confidence. We further extend our previous work [13] 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 6.2, we show that this filtering is essential to achieve attack effectiveness.

[11]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, that is, 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.

3.3 The Attacks

In this article, 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.

3.3.1 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 that is, $\mathcal{R}^f_{v,N,w_1} = \mathcal{R}^f_{v',N,w_2}$.

— The Denoised Reconstructed Profile $\mathcal{R}^f_{v,N,w_1}$ is unique in U_1, and $\mathcal{R}^f_{v',N,w_2}$ is unique in U_2.

Let

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

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

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

Note that, by construction, denoted with:

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

and

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

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

Thus, the probability of *correct* re-identification, that is, 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

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

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

The probability of an incorrect re-identification, that is, a 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, that is, a TP, if $v' = v$. If instead $v' \neq v$, the re-identification is successful but wrong, that is, a FP.

3.3.2 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, that is, $\mathcal{R}^f_{v,N,w_1} \subseteq \mathcal{G}_{v',N,w_2}$ and $\mathcal{R}^f_{v',N,w_2} \subseteq \mathcal{G}_{v,N,w_1}$.

— No other user v'' exists such that $\mathcal{R}^f_{v,N,w_1} \subseteq \mathcal{G}_{v'',N,w_2}$ and $\mathcal{R}^f_{v'',N,w_2} \subseteq \mathcal{G}_{v,N,w_1}$.

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

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

and

$$\widehat{P}_U := Prob\left(\cap_{v' \neq v}\left(\left\{\mathcal{R}^f_{v,N,w_1} \nsubseteq \mathcal{G}_{v',N,w_2}\right\} \cup \left\{\mathcal{R}^f_{v',N,w_2} \nsubseteq \mathcal{G}_{v,N,w_1}\right)\right\}\right) \tag{1}$$

while

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

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 $\mathcal{R}^f_{u,N,w}$ and the Global Reconstructed Profile $\mathcal{G}_{u,N,w}$, 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 FP matches.

3.3.3 Asymmetric Weighted Hamming Attack. For comparison, we consider the **Asymmetric Weighted Hamming Attack** (**AWHA**) introduced in [4]. Authors of [4] 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 FPs as we will discuss in Section 6.1.

4 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**). The dataset is the same that we used in our previous work [13], but we nonetheless report the collection methodology and the characterization for the sake of completeness.

4.1 Data Collection Methodology

In the context of the PIMCity project,[12] we designed, implemented and deployed a fully-fledged online PIMS called EasyPIMS and opened it for experimentation [14]. 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 article, 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 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.

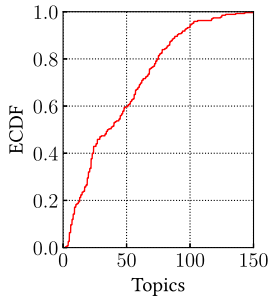
Ethical Aspects. Our data collection process is compliant with ethical principles and EU privacy regulations. EasyPIMS was part of a European Project involving 12 partners and the European Commission has approved all the data collection and processing procedures. Users voluntarily participated, were 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.

4.2 Characterization of Users and Topics

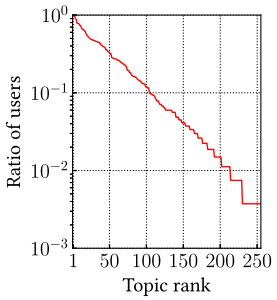
In total, our dataset includes 2,813,283 webpage visits to 50,976 different websites. The number of visits per user per week varies significantly, with some users using the platform for a few weeks and others for the whole four-month experimental period. Some users even installed the plugin on multiple 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

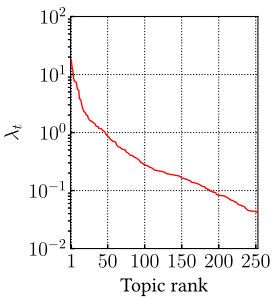
[12] https://www.pimcity-h2020.eu/, accessed on August 16, 2024



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



(b) Ratio of users visiting a topic. Topics in decreasing popularity across the overall population.



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

Fig. 2. Characterization of topic visits.

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 $\mathcal{P}_{u,e}$. We report the ECDF in Figure 2(a). 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 2(b) reports the ratio of users visiting a given topic. The top 5 topics in the population are Search Engines, News, Arts and Entertainment, Internet and Telecom, and Business and Industrial. The most popular topic is visited by 99,3% of users, while up to the top-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 2(c), that is, how many times a topic is visited in an epoch by a user on average. 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

$$\lambda_t = \sum_{u \in U_{|t}} \frac{\lambda_{u,t}}{|U_{|t}|} \tag{2}$$

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 $\mathcal{T}_{u,e}$ depends on the rate of visits $\lambda_{u,t}$ the user u has for the topics t they are interested in during the eth 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, and so on. 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.

5 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.

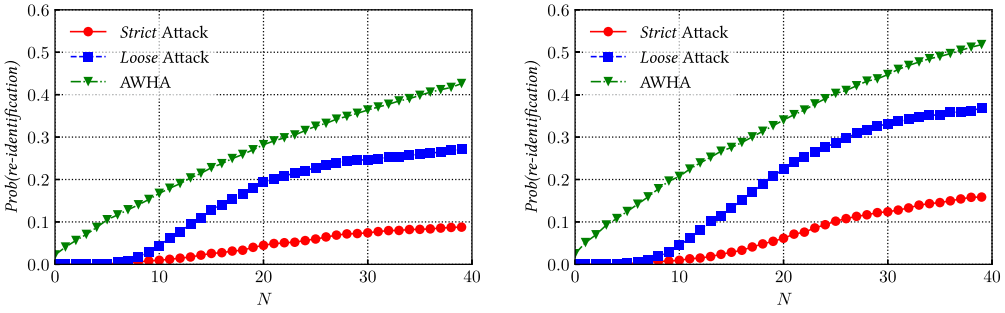
I.I.D. Personas. We create a population of independent and identically distributed (i.i.d.) personas obeying the same marginal statistics as the set of real users from our dataset. To this end, we leverage (i) the marginal ECDF of the number of topics per user (Figure 2(a)), (ii) the marginal empirical distribution of the topic popularity (Figure 2(b)), and (iii) the average empirical rate of visits for each topic λ_t (Figure 2(c)). 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 [5] 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 2(a)).
(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 2(b)).
(3) For each $t \in C_u$, we assign an effective visit rate λ_t from Equation (2), which equals the average empirical visiting rate (Figure 2(c)).

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 [28]. 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.

The persona models above represent pessimistic and optimistic scenarios. For completeness, in the Appendix we introduce two population models that represent the worst-case scenarios and best-case scenarios.



(a) I.I.D. personas.

(b) Crossover personas.

Fig. 3. Probability of a user being correctly re-identified across the epochs, by the means of different attacks.

5.1 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 $\mathcal{T}_{u,e}$, and from it the Profile history $\mathcal{P}_{u,e}$ which contains only the top-z topics (Step 2 of Topic API algorithm). Next, we generate the Exposed Profile $\mathcal{P}_{u,e,w}$, 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 $\mathcal{R}_{u,N,w}$ for each persona $u \in U$ after filtering the Global Reconstructed Profile $\mathcal{G}_{u,N,w}$.

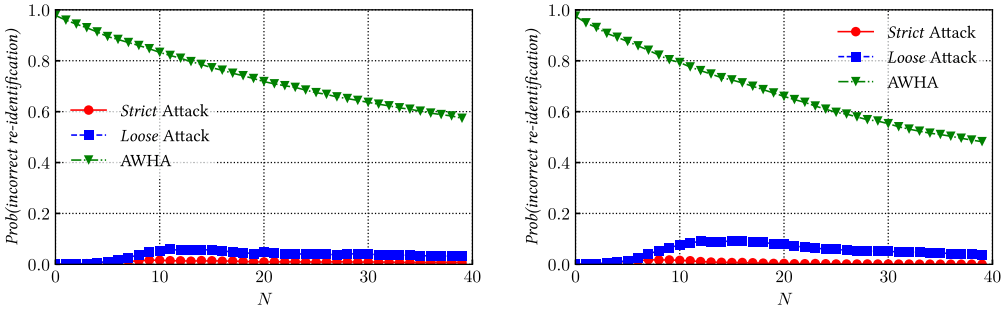
6 Results

In this section, we illustrate the results of our study. We first compare the effectiveness of the different attacks presented in Section 3. 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 considered a population of $|U| = 1,000$ personas. We repeat each experiment 10 times and report the average performance. As introduced in Section 3.1, 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.

6.1 Comparison of Attack Models

We first compare the performance of the three attacks presented in Section 3, and show the results in Figure 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× better performance with respect to the *Strict* Attack (red line), reaching around 25% in *Prob(re-identification)* after $N = 30$ epochs and almost 28% after $N = 40$



(a) I.I.D. personas.

(b) Crossover personas.

Fig. 4. Probability of a user being incorrectly matched across the epochs, with the *Strict* Attack and *Loose* Attack.

epochs, for I.I.D. personas (Figure 3(a)) and almost 38% for Crossover personas (Figure 3(b)). With Crossover personas *Prob(re-identification)* moderately improves, as personas are, by construction, more heterogeneous. As mentioned in Section 3, 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: for example, a topic overcoming the denoising threshold on website w_1, while not being able to do so in w_2. The other attack achieves worse performances, below 10% (20% with Crossover personas). The likelihood-based AWHA proposed by [4] overcomes both the *Strict* Attack and the *Loose* Attack, exceeding 40% (50%) with I.I.D. (Crossover) personas. However, we show thereafter how the higher *Prob(re-identification)* comes with the cost of a large probability of error. The fraction of users incorrectly re-identified could substantially reduce the attack effectiveness, and that we discuss in the following.

In fact, an incorrect re-identification may happen: the attack provides a match for the target user which is incorrect. 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. 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 by design. This does not happen in the *Strict* Attack and *Loose* Attack, where the attack matches no profile if the conditions are not met. As such, the fraction of incorrectly matched users for AWHA largely outnumbers the ones for the other attacks. To reduce incorrect re-identifications, one could set a threshold D_{max} to reject the identification if the distance is higher than D_{max}, introducing a *no-match* option in the AWHA.

In Figure 5, we show how the *Prob(re-identification)* and *Prob(incorrect re-identification)* vary for the AWHA attack when introducing the threshold D_{max}. We consider N=30 epochs. The algorithm finds no match if $D_{max} < 63$. For higher values, both *Prob(re-identification)* and *Prob(incorrect re-identification)* start growing. Yet, *Prob(incorrect re-identification)* grows quicker than *Prob(re-identification)*. This is because there are a lot more possible users that could generate a false-but-closer sequence than the correct-but-looser sequence the victim generates. In fact, by construction, the AWHA algorithm returns the users with the closest distance, that is incorrect with higher probability (i.e., *Prob(incorrect re-identification)* > *Prob(re-identification)*, as shown in Figures 3 and 4).

Thus the benefits introduced by the threshold mechanism are limited and an attacker using a threshold to reduce the amount of FPs will end up reducing the TP to a larger extent.

The *Strict* Attack and *Loose* Attack are more suitable solutions for an attacker to the Topics API: using them, the attacker minimizes the probability of an incorrect match. Conversely, AWHA

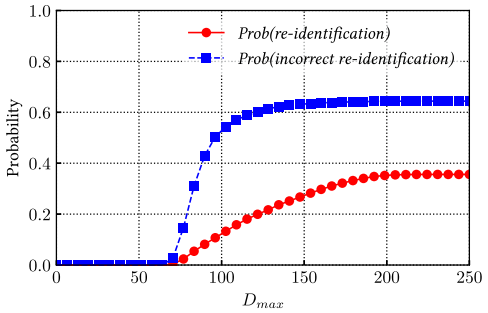


Fig. 5. The impact of a maximum distance threshold D_{max} for the AWHA, N=30.

does not offer such a benefit. Incorrect re-identification could, for instance, push an attacker to define a personalized marketing strategy under the false assumption that a target user has visited two colluding websites. If the rate of incorrect re-identifications exceeds 50%, it means that every decision of the attacker will be incorrect 50% of the times, possibly with severe financial costs. To minimize such cost, the attacker may thus be interested in an attack that limits the amount of incorrect re-identification, although with fewer correct re-identifications.

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.

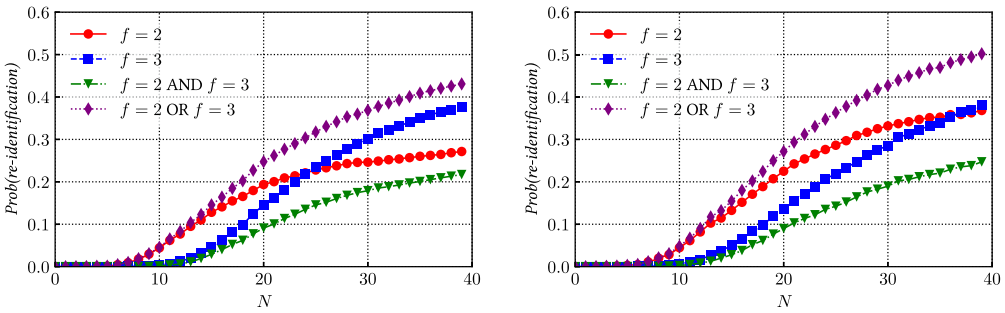
At last, observe that it is not possible to use the classical metrics for classification tasks, such as F1-Score, precision or recall—nor True/FP Rate—because the *Strict* Attack and *Loose* Attack are not binary classifiers, that is, *Prob(re-identification)*+*Prob(incorrect re-identification)* < 1. This happens because a no-match option exists, that is, the one of a user not being matched with any other user.

In the remainder of this Section and in Section 7, we only consider the *Loose* Attack, as it provides the best trade-off between *Prob(re-identification)* and *Prob(incorrect re-identification)*, compared with the other two attacks. We keep comparing the results with both I.I.D. and Crossover personas.

Takeaway: *Under the current threat model, the Topics API still leave a considerable percentage of users at risk of being re-identified. Google's AWHA returns the highest probability of correctly re-identifying the user, but, being so aggressive, it comes with a large portion of incorrect re-identification. From an attacker's perspective, the* Loose *Attack results the best.*

6.2 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 thresholds. Notice that f should increase with epochs N as the attacker has a higher probability of observing multiple times the same random or rare topics. This was already



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

Fig. 6. Probability of a user being correctly re-identified across the epochs, by the means of different threshold rules.

evident in Figure 3(b): 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.

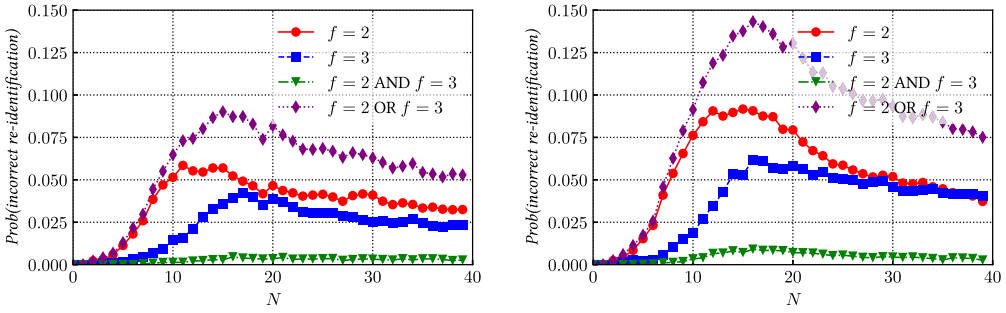
To better understand the impact of f, we show in Figure 6 how *Prob(re-identification)* evolves with $f = 2$ and 3. Later, we also propose a couple of compound strategies. For the sake of readability, we omit to represent the case with $f = 1$: in fact, the *Prob(re-identification)* never exceeds 3% 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 $\mathcal{R}$. However, when the number of epochs increases, the attack becomes less effective, allowing a number of random and rare topics to pollute $\mathcal{R}$. 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 $\mathcal{R}$. 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.

6.2.1 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, due to the filtering threshold a user's profile can be unique[13] 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,

[13]With abuse of language, under the *Loose* Attack, we say that user's profile is unique if the event in r.h.s. of (1) occurs.



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

Fig. 7. Probability of a user being incorrectly re-identified across the epochs, by the means of different threshold rules.

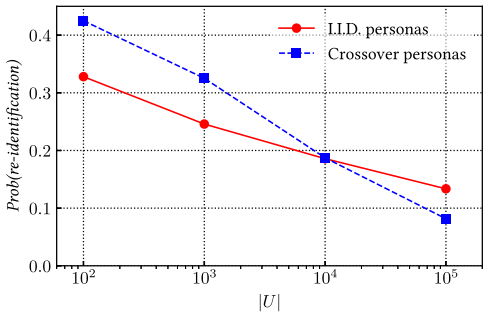


Fig. 8. Probability of being re-identified with different numbers of personas.

labelled as "$f = 2$ AND $f = 3$" in Figure 6), *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 7. 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.

In the analysis, we limit the study to $f \leq 3$. The benefits of $f > 3$ would appear for very large observation windows (i.e., for $N > 40$) which makes the analysis not interesting.

Takeaway: *Different threshold f values impact the efficiency of the attack. Moreover, combinations of threshold can offer lower and upper bounds to the Prob(re-identification) and Prob(incorrect re-identification). Since an attacker cannot know the underlying topic-visiting rate distribution, they cannot know in advance the optimal f value to use. Such bounds can thus allow to understand the expected efficiency range of the attack.*

6.3 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 re-identified. The larger the website's audience, the harder the re-identification is. We illustrate this effect in Figure 8, where we

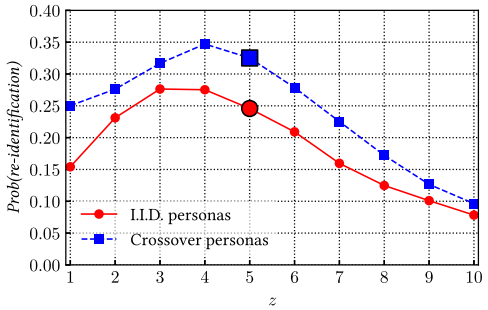


Fig. 9. The probability of an attacker correctly re-identifying a user, with different values of z. $N = 30$, $|U| = 1,000$, $f = 2$, $p = 0.05$. We highlight the *Prob(re-identification)* with the default value $z = 5$.

show how re-identification 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 re-identification candidates. This could enhance the attack even on websites with a large audience, where it would be otherwise easier for the user to *hide in the crowd*. Fingerprinting techniques could help the attacker partition a large number of users into smaller sub-populations, each of which could be the target of an attack independent from the others; the reduction of the population dimension would improve the attack chances, as the total virtual number of users would decrease.

Takeaway: *A large website popularity allows the user to "hide in the crowd". However, techniques exist to partition and reduce the size of the victim audiences, increasing the attacker's re-identification probabilities.*

7 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, that is, the number of topics that are selected every epoch to build the profile $P_{u,e}$ of a user, and p, that is, the probability at which an exposed topic is replaced with a random one. In the following experiments, we consider $N = 30$, $|U| = 1,000$, $f = 2$.

7.1 The Number of Topics in the Profile

In Figure 9, 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

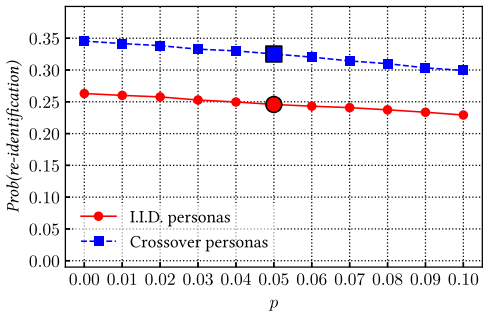


Fig. 10. Probability of a user being re-identified, with different values of p. $N = 30$, $|U| = 1,000$, $f = 2$, $z = 5$. We highlight the *Prob(re-identification)* with the default value $p = 0.05$.

are easier to be linked and thus re-identified. Further increasing 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 $\mathcal{P}_{u,e}$ 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.

Takeaway: *The number of topics z composing the users' profile has a large impact on the Prob(re-identification). Under our persona models, reducing the topic set size to $z = 3, 4$ leads to a higher re-identification probability than the default value $z = 5$.*

7.2 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 10, 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 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.

Takeaway: *The fake topic injection probability has a minor impact on the Prob(re-identification) thanks to the efficiency of the filtering algorithm.*

8 Related work

From the dawn of the Web, behavioural advertising has been a pillar of the ecosystem and entailed the collection of personal information through web tracking. This phenomenon has been the subject of several studies that measured its spread [6, 16] or dug into its technical operation [1, 18, 23]. The implications of web tracking on users' privacy have become more and more debated by the industry [11] and by the research community [9, 15, 24]. It also fostered the birth of anti-tracking tools (i.e., the Ad and Tracker Blockers [19]) and encouraged the legislator to issue privacy-related regulations, such as the US CCPA [3] or the European GDPR [10].

FLoC has been the first public effort by Google to go beyond the classical web tracking based on third-party cookies [21]. 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 [8]. 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 [22]. The attack can exploit browser fingerprint to further improve its effectiveness [2]. FLoC's privacy anonymity properties can be broken in several ways [26]. As a response to the critics towards FLoC, Google retired the proposal and conceived the Topics API, whose functioning we describe in Section 2.

The Topics API exposes users' profiles in terms of topics of interest to the websites and advertising platforms. In this article, we study to what extent users' profiles can be used by an attacker to re-identify the same individual across time or space. Past works already demonstrated that profiling users based on their browsing activity can present severe risks to the privacy of the users [9]. They can be identified with high probability based on the sequence of visited websites [12, 17, 27]. Mitigation such as partitioned storage has been put in place to limit the risk, but ways to bypass them exist [20].

Specifically to the Topics API, the same threat we analyze has been already identified by Epasto et al [7] from Google. The authors carry out an information theory analysis and conclude that the attack is hardly feasible. In this article, we go a step further. Our analyses are not limited to an analytical study on profiles' uniqueness but offer a thorough evaluation using real traffic traces and different user models. To the best of our knowledge, Thomson [25] from Mozilla issued the first independent study on the privacy guarantees of Topics API, elaborating on the conclusions by Epasto et al [7]. He, again, used analytical models and raised severe concerns about the offered privacy guarantees. Recently, Carey et al [4] from Google discussed the privacy implications of the Topics API. They define a theoretical framework to determine re-identification risk and propose the AWHA for re-identification. Differently from us, they do not consider any denoising algorithm. In our article, we compare the AWHA attack with our attack strategies, and we find that, even if AWHA shows the best attack in terms of *Prob(re-identification)*, it suffers from a high probability of incorrect match that makes the attack not practical. With a better balance between *Prob(re-identification)* and *Prob(incorrect re-identification)*, we believe that especially the *Loose* Attack could be a threatening solution.

The present work is an extension of our previous one [13]. In this version, we improved the work in several aspects. (i) In [13], we studied the probability of a user being k-anonymous among the profiles collected by a user. We used this probability as a proxy metric to infer the risk undergone by users and introduced what we now define as *Strict* Attack. Here, we directly focus on the *Prob(re-identification)* and *Prob(incorrect re-identification)*. (ii) We introduce a new *Loose* Attack which results more efficient than the *Strict* Attack we introduced in [13] and we compare results with the attack provided by Google in [4]. (iii) We offer better insights into the impact of the threshold f and

use it to mount even more complex and efficient attacks. (iv) We extend and present a thorough analysis of the impact of the various system and scenario parameters.

9 Discussion and Future Work

We finally discuss the limitations of our model, suggest some possible improvements to the Topics API, and present interesting directions for future work.

9.1 Limitations

The Topics API proposal represents a solid improvement for end-user privacy. Although we showed that they do not prevent an identification attack, they definitively improve the situation from the current "all-allowed" scenarios, balancing privacy with the ability to provide personalised advertisements. Considering the attack and the evaluation we presented in this article, we need to consider limitations in our analysis that could make the attack impractical in real scenarios.

First of all, our results show that the attacker needs 10 to 30 weeks (i.e., up to 6 months) to successfully re-identify the victim with significant probability. If the user visits the colluding websites less than once every epoch (or at least every E week), the time needed to carry the attack increases.

In such a long period of time, the underlying stationarity assumption may be unrealistic. Users' interests may present a seasonal effect. While this shift would be indeed observed by both websites, the filtering algorithm may negatively impact the construction of the Reconstructed Profile. In addition, when considering an *a priori* known victim profile, the shift may harm the re-identification attack.

Considering the evaluation we presented in this work, the I.I.D. and Crossover models are two possible means to generate personas, but they cannot consider the overall factors that impact the users' browsing habits. For instance, they do not model user interests shifting in time—as discussed above—and they might not represent correctly the correlations in the interests that exist in the real world. This is especially true for the I.I.D. method, where the users' topic-visiting rates are sampled independently. For this, we introduce also the best- and worst-scenarios in the Appendix to provide lower and upper bound to the *Prob(re-identification)*.

Related to the above, the original dataset can be improved, both in the size and the representativity of a true population. To this end, we offer our code and tools as open source, which can effectively be reused with larger datasets or more sophisticated population models.

9.2 Improvement of Topics API

To further improve end-user privacy and reduce the ability to perform a re-identification attack, some additional measures could be considered:

- **Limit cookie lifetime:** Periodically expiring the first-party and third-party cookies would bring an immediate privacy-related benefit, as it prevents websites from accumulating information on users for a long time. Figures 3 and 6 show that deleting the first-party cookies every $N = 10$ epochs would keep the *Prob(re-identification)* below 5% with the current attack setup, and *Prob(incorrect re-identification)* would settle to a similar value. This makes the re-identification attack impractical.
- **Topics API parameter optimization:** The suggested values of z, E, and p proposed in the Topics API draft are open to further review. We believe that the code we offer can be used as a tool to investigate how different parameter choices impact the probability of re-identifying users, and choosing the best combination. In particular, we already observed that z has a significant impact in the *Prob(re-identification)*.

— **Probabilistic profile:** The profile of a user $\mathcal{P}_{u,e}$ is currently populated by the z-top topics in epoch e. It could be worth considering other approaches to build the profile (e.g., probabilistic). This will also better balance the utility for the advertisers while not affecting the privacy of the users. This would also benefit the exposure of less-popular topics, opening opportunities to less-mainstream topics/ads to be considered.

9.3 Future Directions

Our work provides a first study on how re-identification attacks can be successful inside the Topics API framework, opening to several angles to be further explored.

First, we rely on a dataset collected from the set of volunteers who participated in the EasyPIMS experimentation. As such, we cannot verify the dataset is representative of general human behaviour. The process of gathering such kind of personal data is cumbersome, but a larger and more heterogeneous audience may help in drawing more solid conclusions. In a similar direction, it is interesting to evaluate more diverse population generation approaches, including diverse usage patterns, classes of users, and so on.

Second, the threat model could be improved: for instance, the attacker could consider more sophisticated techniques to match the profiles. It may be interesting to scale the attack surface considering more than two colluding websites.

Third, we suppose the attacker has no background knowledge of the victims. As said above, this assumption can be relaxed to study to what extent any additional information on the user (e.g., retrieved through the IP address or browser fingerprinting techniques) can help the attacker.

Fourth, in this work we did not consider the semantics and possibly correlations among the topics in the taxonomy. However, they are relevant both for a business-centered and a privacy-centered analysis. At the moment of writing, Google announced a new taxonomy[14] with more topics—but did not offer the code to map websites to it. Moreover, the accuracy of the mapping model, which works only on hostnames, is up for debate. Future work could explore these lines of research.

Finally, since at the moment of writing Google started giving access to the Topics API to real users and advertisers, one could try to mount the attack in a real-world scenario.

The Topics API is a novel proposal by Google, and we believe the research community should further work to understand the implications of its design, as it might become the *de facto* standard for online advertising in the near future.

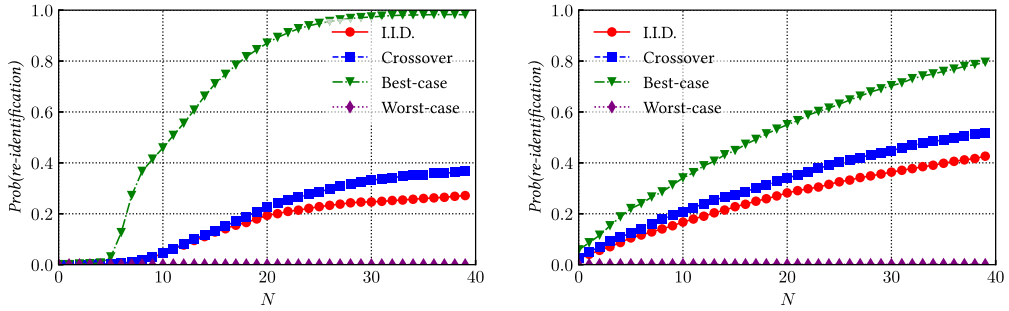
Appendix

A Attacker's Best- and Worse-Case Scenarios

While introducing the two population models we presented in this article, I.I.D. and Crossover (see Section 5), we mentioned the fact that some models could be a *more optimistic* scenario for an attacker. In particular, we stated that Crossover personas have more heterogeneous profiles, and thus it is easier for an attacker to tell personas apart based on their reconstructed profiles. Our results (e.g., Figure 3) confirm our intuition.

In this appendix, we go in the direction of visually quantifying how much the I.I.D. and Crossover methods are optimistic from an attacker's point of view. In order to do so, we compare them with the best- and worst-case users' scenarios for the attacker. To define these scenarios, we focused on the *Strict* Attack and *Loose* Attack.

[14] https://github.com/patcg-individual-drafts/topics/blob/main/taxonomy_v2.md, accessed on August 16, 2024.



(a) *Loose* Attack. (b) AWHA.

Fig. 11. Probability of a user being correctly re-identified across the epochs, comparing different possible user models.

We consider a worst-case scenario for an attacker a situation where the users' profiles are indistinguishable with the higher possible probability. This happens when every user exposes exactly z topics in a week, and the set of exposed topics is the same for every user. To obtain such scenario, in our experiment we set $z = 5$ topics with $\lambda = 5\log(10)$, and we set the λ for every other topic to zero. We choose this value of λ such that the probability of a user not visiting the topic is at most 10^{-5}. Essentially, we force every user to exhibit the same profile $\mathcal{P}_{u,e}$ In short, in this scenario, every user will certainly visit $z = 5$ and $z = 5$ only topics, which will thus populate their profile without the risk of introducing padding topics in the user's profile (see Section 2 Step 2).

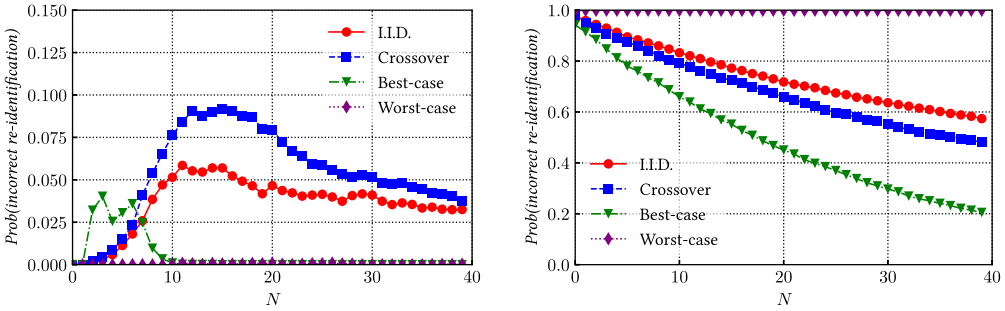
On the other side, a best-case scenario for an attacker is a setting that maximises the probability of profiles being different with any other user. We obtained such setting by assigning to every user a 5-tuple of topics with $\lambda = 5\log(10)$, being careful that no users share the same 5-tuple. All the other topics' λs are set to zero. In this way, after a sufficient amount of time, the two websites will have collected all the topics in the 5-tuple of each users, making it trivial to correctly match the users—as every 5-tuple is unique to a user in this scenario.

We show the results of our analyses in Figure 11. In particular, Figure 11(a) we show the impact of the user model on the *Prob(re-identification)* presuming a *Loose* Attack. We observe that in the worst-case scenario no user is re-identified: this happens because we keep close to zero the probability of a user's profile being unique on two websites, and thus to be matched across websites. On the other hand, once every topic has been exposed more than the threshold f times, the *Prob(re-identification)* with the best-case scenario rapidly approaches 1: every user is re-identified, thanks to the orthogonal sets of exposed topics which are not polluted by random topics.

In Figure 11(b), we observe the impact of users model for the AWHA. Here, in the best-case scenario we observe that the *Prob(re-identification)* goes to 1 more slowly: this happens because, although orthogonal in their set, individual topics can appear in multiple profiles, and their temporal co-occurrence reduces the chances of the AWHA to correctly match personas.

With both attacks, we can observe that I.I.D. and Crossover personas are, generally speaking, midway between the fully-orthogonal and the fully-homogeneous user scenarios, with the Crossover model slightly closer to the former. The level of correlation in the dataset from which the two models derive their personas moves them away from the best-case scenario, while not forcing equal visiting rates on every topic.

Finally, in Figure 12 we observe the probability of incorrectly matching users in the worst- and best-case scenarios. The AWHA case is less interesting, as the *Prob(incorrect re-identification)* can be evaluated as 1−*Prob(re-identification)*. On the other hand, an attacker exploiting a *Loose* Attack



(a) *Loose* Attack. (b) AWHA.

Fig. 12. Probability of a user being *incorrectly* re-identified across the epochs, comparing different possible user models.

would not observe any incorrectly matched users after a few epochs. In the worst-case scenario, since all the users have the same profile, the attacker cannot attempt any match. In the best-case scenario, since a unique 5-tuple corresponds to each user, in principle, users cannot be incorrectly matched provided that the 5-tuples of users have been correctly identified. However, a few incorrect matches may occur after a few epochs, when it is possible that some topic in the 5-tuple has not been yet exposed, and profiles of different users can temporarily overlap.

References

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

[2] Alex Berke and Dan Calacci. 2022. 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* . Association for Computing Machinery, New York, NY, USA, 337–349.

[3] California State Legislature. 2018. California Consumer Privacy Act of 2018. Retrieved from https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375 (Last accessed September 6, 2021).

[4] C. J. Carey, Travis Dick, Alessandro Epasto, Adel Javanmard, Josh Karlin, Shankar Kumar, Andres Muñoz Medina, Vahab Mirrokni, Gabriel Henrique Nunes, Sergei Vassilvitskii, and Peilin Zhong. 2023. Measuring Re-identification risk. *Proceedings of the ACM on Management of Data* 1, 2, Article 149 (june 2023), 26 pages. DOI: https://doi.org/10.1145/3589294

[5] Luc Devroye. 1986. Sample-based non-uniform random variate generation. In *Proceedings of the 18th Conference on Winter Simulation*. 260–265.

[6] Steven Englehardt and Arvind Narayanan. 2016. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1388–1401.

[7] Alessandro Epasto, Andres Munoz Medina, Christina Ilvento, and Josh Karlin. 2022. Measures of Cross-Site Re-Identification Risk: An Analysis of the Topics API Proposal. Retrieved from https://github.com/patcg-individual-drafts/topics/blob/main/topics_analysis.pdf (Last accessed February 27, 2023).

[8] 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. 2021. Clustering for private interest-based advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* . Association for Computing Machinery, New York, NY, USA, 2802–2810.

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

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

[11] Stephen Farrell and Hannes Tschofenig. 2014. Pervasive Monitoring Is an Attack. RFC 7258. DOI: https://doi.org/10.17487/RFC7258
[12] Dominik Herrmann, Christian Banse, and Hannes Federrath. 2013. Behavior-based tracking: Exploiting characteristic patterns in DNS traffic. *Computers & Security* 39, part A (2013), 17–33.
[13] Nikhil Jha, Martino Trevisan, Emilio Leonardi, and Marco Mellia. 2023. On the robustness of topics API to a Re-identification attack. In *Proceedings of the on Privacy Enhancing Technologies 2023(4)*. 66–78.
[14] 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. 2022. A PIMS development kit for new personal data platforms. *IEEE Internet Computing* 26, 3 (2022), 79–84.
[15] Jonathan R. Mayer and John C. Mitchell. 2012. Third-party web tracking: Policy and technology. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*. IEEE, 413–427.
[16] Hassan Metwalley, Stefano Traverso, Marco Mellia, Stanislav Miskovic, and Mario Baldi. 2015. The online tracking horde: A view from passive measurements. In *Proceedings of the International Workshop on Traffic Monitoring and Analysis*. Springer, 111–125.
[17] Lukasz Olejnik, Claude Castelluccia, and Artur Janc. 2012. Why johnny can't browse in peace: On the uniqueness of web browsing history patterns. In *Proceedings of the 5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*. Spain.
[18] Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. 2021. *User Tracking in the Post-Cookie Era: How Websites Bypass GDPR Consent to Track Users*. Association for Computing Machinery, New York, NY, USA, 2130–2141.
[19] Enric Pujol, Oliver Hohlfeld, and Anja Feldmann. 2015. Annoyed users: Ads and ad-block usage in the wild. In *Proceedings of the 2015 Internet Measurement Conference*. 93–106.
[20] Audrey Randall, Peter Snyder, Alisha Ukani, Alex C. Snoeren, Geoffrey M. Voelker, Stefan Savage, and Aaron Schulman. 2022. Measuring UID smuggling in the wild. In *Proceedings of the 22nd ACM Internet Measurement Conference* (Nice, France). Association for Computing Machinery, New York, NY, USA, 230–243.
[21] Deepak Ravichandran and S Vasilvitskii. 2021. Evaluation of cohort algorithms for the FLoC API. Retrieved from https://github.com/google/ads-privacy/raw/master/proposals/FLoC/FLOC-Whitepaper-Google.pdf (Last accessed February 27, 2023). *Google Research & Ads white paper* (2021).
[22] Eric Rescorla and Martin Thomson. 2021. Technical comments on FLoC privacy. Retrieved from https://mozilla.github.io/ppa-docs/floc_report.pdf (Last accessed February 27, 2023). (2021).
[23] Valentino Rizzo, Stefano Traverso, and Marco Mellia. 2021. Unveiling web fingerprinting in the wild via code mining and machine learning. *Proceedings on Privacy Enhancing Technologies* 2021, 1 (2021), 43–63.
[24] Janice C. Sipior, Burke T. Ward, and Ruben A. Mendoza. 2011. Online privacy concerns associated with cookies, flash cookies, and web beacons. *Journal of Internet Commerce* 10, 1 (2011), 1–16.
[25] Martin Thomson. 2023. A privacy analysis of google's topics proposal. Retrieved from https://mozilla.github.io/ppa-docs/topics.pdf(Last accessed February 27, 2023). (2023).
[26] Florian Turati. 2022. *Analysing and Exploiting Google's FLoC Advertising Proposal*. Master's thesis. ETH Zurich, Department of Computer Science.
[27] Luca Vassio, Danilo Giordano, Martino Trevisan, Marco Mellia, and Ana Paula Couto da Silva. 2017. Users' fingerprinting techniques from TCP traffic. In *Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*. 49–54.
[28] Bimal Viswanath, M. Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. 2014. Towards detecting anomalous user behavior in online social networks. In *Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14)*. 223–238.

Received 27 December 2023; revised 17 April 2024; accepted 9 June 2024