A complementary utility and privacy trade-off evaluation of Google's FloC API

Guillaume Kessibi, Aymen Ould Hamouda, Charly Poirier, Antoine Boutet

A complementary utility and privacy trade-off evaluation of Google's FloC API

Guillaume Kessibi
INSA Lyon
Lyon, France
guillaume.kessibi@insa-lyon.fr

Aymen Ould Hamouda
INSA Lyon
Lyon, France
aymen.ould-hamouda@insa-lyon.fr

Charly Poirier
INSA Lyon
Lyon, France
charly.poirier@insa-lyon.fr

Antoine Boutet
Univ Lyon, INSA Lyon, Inria, CITI
Lyon, France
antoine.boutet@insa-lyon.fr

Abstract

Controversial online advertising practices such as hidden monitoring of users activities from unknown companies through third-party cookies (also known as tracking cookies) has raised concerns of users about the privacy of their personal data. To address these concerns, Google proposed an alternative to third-party cookies called FLoC (Federated Learning of Cohorts) API, designed to provide a privacy-preserving mechanism for interest-based ad selection. The underlying idea of FLoC is simple: use a cohort based approach (i.e., a cohort is a group of users ranging in the thousands with the same interests) instead of tracking users individually on the internet. Google has shown the valuable of Floc for interest based advertising through evaluation using datasets (e.g., MovieLens), and has conducted a trial in Spring 2021 to assess FLoC in real conditions. However, the utility and privacy trade-off of FLoC is still unclear and several comments have pointed out limitations of the technology. In this paper, we conducted complementary experimental evaluations to better understand the utility and privacy trade-off of FLoC. Specifically, we evaluated the behavior of cohorts in terms of user acquaintance and churn (driving the utility), and the uniqueness of cohorts path of users (to assess the privacy). We show that the valuable performances obtained from simulations using MovieLens dataset do not provide the same conclusions as the observations done during the real life experiment. This substantial performance difference coming from the exploitation of data with a different nature (i.e., a sparse dataset of movie ratings versus the browsing history of the users), questions the choice of the considered dataset to evaluate a solution.

Keywords: datasets, neural networks, gaze detection, text tagging

1 Introduction

Online advertisements are everywhere on the Internet, they follow you when you search for information, read news, check social media, or buy items. Online advertising relies on tracking user visits and their associated interactions to target ads. The intrusive nature of this tracking raises many privacy issues starting to question more and more users. The growing awareness of the population about privacy issues leads to an increasing usage of ad blockers[1] for instance. In addition, stronger regulations on data protection (e.g., GDPR, HIPAA) have been enforced and new services making privacy an incentive vector such as privacy-based search engine (e.g., Duckduckgo, Qwant), web browsing (e.g., Web Proxy, Tor, Brave), or mailing (e.g., Protonmail) emerge. To answer this desire of more privacy, the online advertising eco-system tries to evolve with different practices. Such new practices include the usage of dark patterns to obtain user consent through trick questions and visual interference [13], or initiatives such as the Privacy Sandbox. This initiative from Google aims to create technologies that both protect people's privacy online and give companies and developers tools to build thriving digital businesses. More specifically, the Privacy Sandbox aims to reduce cross-site and cross-app tracking (e.g., through third-party cookies) while still monetizing user visits through targeted ads.

As part of this Privacy Sandbox initiative, Google announced the upcoming release of the Federated Learning of Cohorts API (or FLoC API) documented in a white paper [18]. The underlying idea of FLoC is simple: use a cohorts based approach instead of tracking users individually on the internet. A cohort represents a group of users ranging in the thousands with the same interests. Each cohort is updated once every week in order to better match users that belong to according to the evolution of their interests. In addition to their simulation-based experiments [18], Google has also launched a trial in Spring 2021 to assess FLoC in real conditions. Google advertised FLoC as respectful of the user privacy as profiles are not built directly from activity online of each individual but from the aggregated activity of a group of users with similar interests (i.e., ensuring k-anonymity).

[1] https://backlinko.com/ad-blockers-users

Moreover, Google promised that FLoC API is a viable solution for marketers to have effective digital advertising across the Web.

Although the assessment of the utility of FLoC has been reported in Google's white paper through datasets, the impact on privacy was not clear and several comments have pointed out limitations [17, 20]. Early 2022, Google has finally given up the project and announced a new proposal called Topics.

Besides FLoC is dead, this paper investigates this proposal with regards to utility-privacy trade-off. More precisely, we analyze the behavior of cohorts in terms of user acquaintance and churn (driving the utility) and the uniqueness of cohorts path of users (to assess the privacy) using MovieLens dataset. We show that this dataset is certainly not relevant for evaluating FloC as the results are drastically different to ones observed during the origin trial realized by Google. Indeed, results using MovieLens are convincing to show the value of Floc for interest based advertising, however observations from the real life experiment do not conduct to the same conclusion. This substantial performance difference coming from the exploitation of data with a different nature (i.e., a sparse dataset of movie rating versus the browsing history of user), questions the choice of the considered dataset to evaluate a solution.

This paper is organized as follows: Section 2 introduces the background of FLoC and state-of-the-art, Section 3 details our evaluation and presents our results. Finally, Section 4 concludes.

2 Background and state-of-the-art

FLoC was announced about a year ago, in January 2021 [2]. Beside FLoC API was still in its early days, the ambition of Google was to deploy it in 2023. However, Google has give up the project one year after its announcement, in January 2022 [3]. In this section, we first present the background of FLoC API (Section 2.1) before presenting comments and different point of views on the limitation of the technology shared in the scientific community (Section 2.2).

2.1 Federated Learning of Cohorts (FLoC)

Before describing the system, it is important to notice that some key points of its implementation have changed since its first announcement, however source codes of FLoC are not publicly available. Google has performed a trial through their Chrome browser which took place in spring 2021. The following description is based on the last Google's research paper [8] on FLoC.

FLoC aims to replace cookies for interest-based advertising by instead grouping users into groups of users with comparable interests called cohort. To achieve that, FLoC relies on interest-based identifiers called *cohortId* built locally on each user from their browsing history. The minimum size of a cohort is fixed at 2,000 users to ensure a certain level of indistinguishability. To ensure this minimum size, FLoC needs the support of a central server.

The FLoC API is described to work as follows:

1. The browser *locally* prepares the browsing history to keep only relevant information for the computation of cohorts (i.e., domain names visited on a regular basis). This produces a vector of browsing interests for each user.
2. The browser hashes the stripped history of visited domains using SimHash in combination with either SortingLSH or PrefixLSH to generate a cohortId with a specific length. This identifier is constructed in such a way that users with a close interest vector are more likely to have the same or at least close cohortId.
3. While browsing, websites and advertising companies can retrieve a user's cohortId[4] by calling a simple JavaScript function in their site source code.
4. Ad tech companies then learn that a given cohortId spends time on specific websites. These companies will then target ads according to both websites and their associated content, and users sharing the same cohortId.
5. The process is restarted every 7 days to take into account the change of interests of users.

SimHash is a simple hashing algorithm focused on preserving the locality of the data initially developed to identify near duplicate documents [4, 12, 21]. SimHash takes a vector x of size d as input and outputs a p-bit vector $Hp(x) \in \{0,1\}^p$ which represents the hash of x. A simple example[5] is depicted Figure 1. Equation 1 shows how to compute the i^{th} bit of the p-bit hash vector, $w_1, ..., w_p$ being different d-sized random unit-norm vectors.

$$H_p(x)[i] = \begin{cases} 0 & \text{if } w_i \cdot x \leq 0 \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

FLoC leverages SimHash for its propensity to build similar hashes associated with similar inputs (i.e., browsing history). More precisely, the probability of mapping two vectors x_1 and x_2 to the same p-bit cohortId depends on the angle between them, input vectors with small angles are more likely to share the same cohort than vectors with large angles.

[2] https://blog.google/products/ads-commerce/2021-01-privacy-sandbox/

[3] https://techcrunch.com/2022/01/25/google-kills-off-floc-replaces-it-with-topics/

[4] https://web.dev/floc/#how-does-the-floc-javascript-api-work

[5] This was run using hybridtheory/floc-simhash's implementation and README example

```
input:
  "google.com|hybridtheory.com|youtube.com|reddit.com"
  "google.com|youtube.com|reddit.com"
  "github.com"
  "google.com|github.com"
output:
  ['0x60' '0x60' '0x30' '0x34']
```

Figure 1. Example of SimHash run on $p = 8$ bits

It is important to notice that the computation of SimHash uses a pseudo-random number generation, which means there is not any way to reverse the algorithm in order to get the user's interest vector from its cohortId. In other words, it is impossible to infer interesting personal information about a user using only their cohortId.

However, with this algorithm we have no control over the minimal number of users per cohort. We can increase or reduce the mean number of users per cohort by increasing or decreasing the number of bits in the hash but we cannot ensure that any cohort can have less than k users. As described in [8], two solutions exist to manage this issue. The first one is to gather the cohortIds of all users to a centralized server which deactivates (i.e., make unavailable for advertisers) all the cohorts which are too small. Another option is to bring in the same cohort users from nearby cohorts which are too small to create cohorts which ensure k-anonymity. To do so it is possible to use an algorithm called SortingLSH to homogenize the size of the cohorts. SortingLSH is a centralized method that post-processes the hashes generated by SimHash to ensure k-anonymity. Let $h_i = H_p(x_i)$ the hash generated by SimHash for user i and $\mathcal{H} = \{h_i, ..., h_n\}$ be the multiset of hashes of all users. SortingLSH generates cohorts as follows:

1. Sort $h_1 = H_p(x_1), ..., h_n = H_p(x_n)$ in lexicographical order to get a sorted list of hashes $h_{(1)} \leq ... \leq h_{(n)}$.
2. Assign the sorted hashes to cohorts by partitioning the order in contiguous groups of at least k hashes.

The ordering step ensures that contiguous hashes in this order correspond to users with mostly similar SimHash values. The partitioning follows a recursive approach called PrefixLSH [8]. This recursive approach iterativelly divides the hash string interval (i.e., into both a left interval with bit 0, and a right interval with bit 1). Starting from level 1 with all users in the same cohort, this iterative approach splits the interval until both sides of the string intervals does not contain at least k users.

2.2 FLoC: a controversial project

Although the goal of FLoC to improve privacy is laudable, this project is controversial due to an unclear utility and privacy tradeoff. For instance, Mozilla's Research lab [17] has discussed the privacy risks of FLoC and point out the following issues:

1. *Cross-site tracking*. A website which is able to link the visits of an individual (i.e., through a login account) over a long period of time can draw up the evolution of the cohortID of this user. Since there should be a lot of cohorts compared to the number of users per cohort, it is very likely that a user's *path* of cohortIds will become unique after a certain number of cohort epochs. Consequently, a company observing visits with the same *path* of cohorts through different websites could establish that it is the same user. This cross-site tracking is exactly the problems of third-party cookies and what FLoC is supposed to avoid. This tracking is even made easier if we consider the browser fingerprinting [11, 15]. Indeed, the cohortID path combined to this fingerprinting (e.g., language, name of the browser, version of the browser, OS) can be used to increase the uniqueness and then improve the cross-site tracking of users. Interestingly enough, Google advises to leverage the userAgent associated with the cohortID [18]. We evaluated specifically the uniqueness of the cohortID path combined or not with fingerprinting information in Section 3.
2. *Profiling*. Websites sharing data they have about each cohort can result to make a map of the interests of each cohortId. As the system is designed to allow advertisers to easily compute the interests of each cohort in order to offer targeted advertising, the interests of a user could be easily inferred by getting its cohortID. Moreover, if you know different cohortIDs belonging to someone over time, you can search the intersection of interests between their different cohorts and this intersection might describe someone quite accurately and be used for profiling.

Another interesting point of view on FLoC is described in web articles written by Criteo, a web advertiser which was able to observe this system during the FLoC's origin trial in Spring 2021 [20]. These articles describe what they learnt from the FLoC system from an advertiser's viewpoint.

First, Criteo mentions that the FLoC origin trial involved only a very small portion of Chrome's users. It is also mentioned that the cohorts are built using PrefixLSH, an algorithm which is quite similar to SortingLSH as described in Section 2.1. They observed 33,105 different cohortIds during the origin trial. Criteo was quite disappointed by this origin trial since it only involved a small fraction of Chrome's users making it hard to conclude on how the system might adapt when scaling.

Second, they noticed that the cohortId is recomputed every 7-days for most users but it might be slightly less (6 days) or more (up to 10 days) for some others. Another very interesting result is the measure of the churn rate which is the amount of users who get a different cohortId after a certain period of time. What they observed after 7 days is that 88% of users had a different cohortId. In some cases, it reached 96% after 10 days. They concluded that cohorts were very unstable during the origin trial which might have to do with the amount of bits prescribed in the cohortIds. To dig deeper into stability measures, they also found out that the average cohortId change of users which is equal to 9,600. It means that on cohortIds ranging from 0 to 36,000, after 7 days, the average difference between a user's old cohortId and new cohortId is 9,600. A smaller difference was expected: the smaller it is, the more stable cohorts are, which is not the case here. Their conclusion is that FLoC will be very hard to exploit for interest based advertising since people move a lot between cohorts which makes it really hard to capture one cohort's interests over time.

Third, they computed the similarity of the cohorts over time and they realised that after 6 weeks, the average cohort similarity (i.e., the average cosine over the users of the same cohort, ranging from 0 to 1) goes down to 0.50, which is the absolute minimum to make cohorts at least a little bit useful for interest-based advertising. However, this number goes up on the biggest cohorts (more than 100 identified users) since it remains over 0.75 after 6 weeks. Then they tried to make inferences about what they can learn about the evolution of a cohort's interests over time and they observed that it might remain stable on the biggest cohorts but not on the smaller ones.

Their conclusion is that the origin trial of FLoC was not convincing to show the value but convinced that is perhaps to the small number of users involved in this trial. In the next section, we show that these observations using real browsing histories can be drastically different in case of using dataset such as MovieLens, questioning the validity of the performance evaluation when the considered data are not adapted.

3 Evaluation

To get a better understanding of how the FLoC could impact users' privacy, we implemented the SimHash and the SortingLSH algorithms to build cohorts and conducted different experiments. As no dataset of user browser history is publicly available, we exploited the same dataset used by Google in their white paper: the MovieLens dataset. We first describe our evaluation setup to create the user's interests vector and for computing the cohortId (Section 3.1) before to deeper analyze the FLoC's Cohort IDs (Section 3.2) and the uniqueness of (Section 3.3).

3.1 Evaluation setup

We describe in this section our experiment setup including the considered dataset and the methodology to create the user's interests vector and to compute the cohortId.

3.1.1 The Movielens dataset.

MovieLens [9] is a non-commercial web-based movie recommendation system. It was launched by GroupLens in 1997. Today, they provide a handful of datasets comprising millions of movie ratings. In our study, we used the MovieLens 25M dataset, which is made of 25 million movie ratings from 162,541 users, over a 25-year period. User ratings are quite sparse, as it spans over more than two decades of MovieLens, each user has 153 rating entries on average. Ratings range from 1 to 5 and are paired with a timestamp (Table 1). In addition, the movies are attached to categories (Table 2).

userId	movieId	rating	timestamp
1	1	4.0	964982703
1	3	4.0	964981247
1	6	4.0	964982224
⋮	⋮	⋮	⋮

Table 1. Sample data representative of ratings.csv

movieId	title	genres
1	Toy Story (1995)	Animation, Children
2	Jumanji (1995)	Adventure
3	Grumpier Old Men (1995)	Comedy, Romance
⋮	⋮	⋮

Table 2. Sample data representative of movies.csv

3.1.2 Creating the user's interests vector.

In real life, users' interest vectors would be based on their browsing history. With the MovieLens dataset, we need to create the user interests vectors from the ratings they gave to the movies they saw. To achieve that, the idea is that each dimension of the user's interest vector will correspond to a possible category like *Romance* or *Adventure* for instance.

For each user, we took all the movies they rated for the considered period of time and we created a vector out of each movie. Each of the dimensions of this vector corresponds to a possible movie category. For each category attached to the movie, the related value is set to the rating the user gave to the movie. The value related to other categories (i.e., not attached to the movie) is set to 0. This provides several vectors for each user (i.e., one rating corresponds to one vector). We then computed the average of all those vectors and normalized the resulting vector. The resulting vectors represent the users' interests vectors. Once we have generated all vectors, it is necessary to center all vectors to 0 as the pseudo-random vectors are centered around 0.

3.1.3 Computing the CohortId of each user.

Once we have generated each user's interests vectors, we compute their cohortIds. To do so, we execute the SimHash algorithm to build the hash of all users' interests vectors with different bit lengths. We then compute the SortingLSH on these hashes in order to gather cohorts that would be too small.

3.2 FLoC's Cohort IDs

To better understand the formation of cohorts, we first evaluate how close in terms of interests users belong to the same cohort. To achieve that, we compute the interest vector associated with a cohort (by aggregating interest vectors of users belonging to it) and measure the average similarity between the interest vector of each user of the cohort with the interest vector of the cohort. This cohort similarity is computed as follow:

$$CohortSimilarity(C) = \frac{1}{|C|} \sum_{u \in C} = cos(\vec{u}, \vec{v}), \tag{2}$$

where C is the cohort, $\vec{v}$ is interest vector of the cohort, $\vec{u}$ is the interest vector of user u, and $cos(\vec{u}, \vec{v})$ defined as follow:

$$cos(\vec{u}, \vec{v}) = \frac{\vec{u}.\vec{v}}{||\vec{u}||.||\vec{v}||} \tag{3}$$

A user with a cosine close to 1 will be very close to their cohort, meaning that they have the same interests as most of the users in the cohort. In other words, if their cosine is close to 1, the SimHash algorithm puts them in the good cohort.

If a cohort has a mean cohort similarity close to 1 it means that all the users in this cohort have close interests, which makes it relevant for interest-based advertising. On the other hand, a cohort with a *CohortSimilarity* close to 0 would be quite harmful for interest based advertising because it would mean that the users in this cohort have very different interests. To have a comparison basis, we calculated the cohort similarity of users in a 20-bits SimHash distribution, then of the users in a k-means distribution and the users in a random distribution. For all of them we calculated the mean of all cohort similarities for each year from 1995 to 2019.

The evolution of the cohort similarity over time where the cohort is built through a 20-bits SimHash, a K-means and random users is depicted Figure 2. Results show that the k-means distribution gives a slightly better cohort similarity than SimHash. However, the SimHash has a much higher cosine similarity than the random distribution. This is not surprising that k-means has a better similarity as it brings closest users together considering all the users in the dataset, where the SimHash algorithm computes a cohort for a user without knowing anything about the other users. This shows that the SimHash algorithm is quite good for interest-based advertising while being respectful of privacy since it does not need to centralize users' data to compute cohortIds.

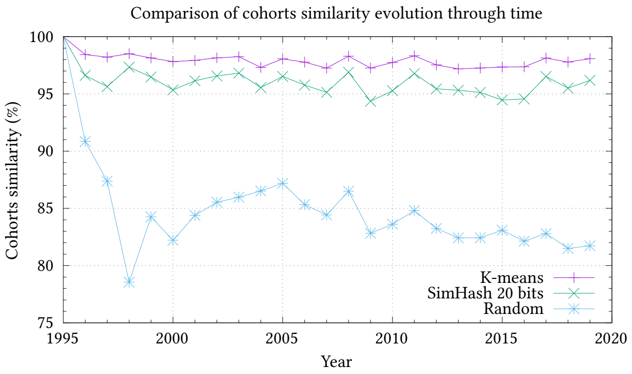


Figure 2. Comparison of cohorts similarity evolution over time (MovieLens dataset): cohorts construction through SimHash gather users with close interests.

Second, we analyze the stability of users constituting a cohort. To do so, we compute cohort distributions using SimHash and SortingLSH and we count the number of users who stay in the same cohort within two-year periods, from 1995 to 2019 (Figure 3).

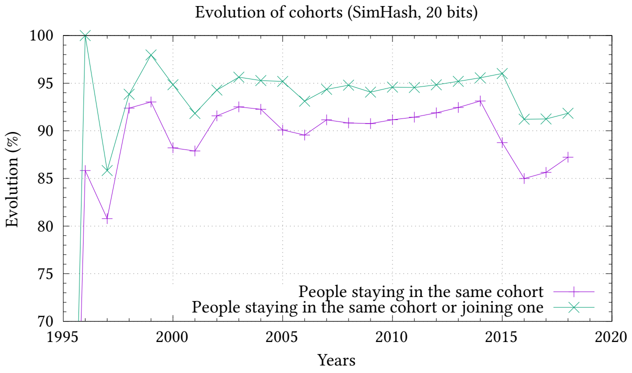


Figure 3. Percentage of people staying in the same cohort from one year to another, cohorts are quite stable over time (MovieLens dataset).

Results depicted Figure 3 show that this number is quite high (about 90% for most years), meaning that most users are staying in the same cohort between years. This means that users are very likely to remain in the same cohort over long periods of time, meaning that it is possible to understand one cohort's interests over a long period of time since it will contain the same users. However, it seems that this fact is only verified in the MovieLens dataset and not in the users browsing habits. As explained in section 2.3, Criteo explains in an article about FLoC's origin trial that in real life conditions, users are changing cohorts very often. They observed that after 7 days, only 12% of the users stayed in the same cohort (and this number goes down to less than 5% after

10 days). This raises an important question about the validity of MovieLens (and similar datasets) for FLoC observations, since a lot of people might always vote for the same type of movies (and a short amount of movies each year) while they visit hundreds of different websites everyday on a very wide range of topics. So, MovieLens is not that great to simulate internet user's browsing habits evolution, and certainly not relevant to test the performance of FloC.

The number of cohorts and the number of users per cohort depends on the parameters of the SimHash and the SortingLSH algorithms. These numbers impact the behavior of those cohorts (e.g., the cohort similarity). Figure 4, Figure 5, and Figure 6 depicted respectively the number of cohorts, the evolution of the cohorts similarity, and the average number of users per cohort over time using SortingLSH with different bit sizes.

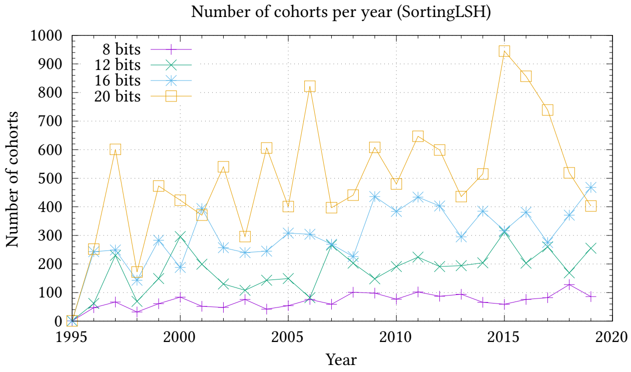


Figure 4. Number of cohorts per year depends on the bit size of the hash (MovieLens dataset).

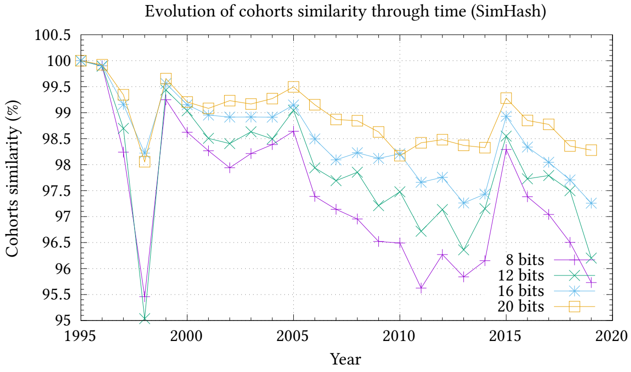


Figure 5. Evolution of cohorts similarity through time: smaller cohorts (through a hash with a larger bit size) gather users with closer interests (MovieLens dataset).

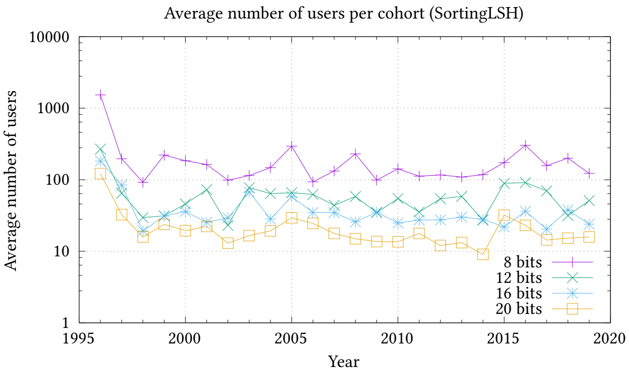


Figure 6. Average number of users per cohort per year: the size of the cohort depends on the bit size of the hash (MovieLens dataset).

As shown in Figure 6, the number of users per cohort is much smaller than what Google wants to do (at least 2.000 users per cohort). It is another artifact related to the MovieLens dataset where many users have not rated a single movie within a year.

In 2021, there were an estimated 3.2 billion Chrome users in the world. That year, Google launched their FLoC Origin Trial, reaching roughly 0.2% of Chrome users[6], representing around 6,400,000 users. Google decided to group them in cohorts with a k-anonymity of 2.000. If we scale this down to our case study with 162.541 users, this means our k-anonymity has to be approximately 5 users per cohort.

3.3 Uniqueness

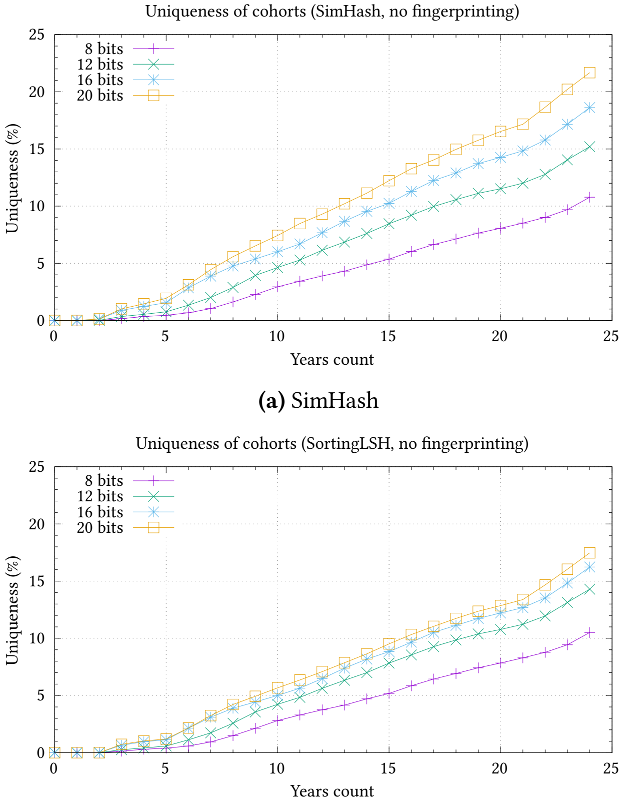
Uniqueness has been observed in various types of user traces that may act as digital fingerprints leading to re-identification such as the personalised configurations of mobile devices [1] or Web browsers [6], the logs of in-car sensors [7], credit card records [14], neighborhoods in Social Networks [19], the writing style of users on the Web [16], or the mobility patterns [3, 5].

Compared to user tracking using third-party cookies, FLoC ensures through PrefixLSH and SortingLSH that users are never alone in a cohort (i.e., unique). However, with the cohortId computation of a user being done every week, a website has the possibility to store a history of old cohortIds of its users. Doing so, the web site can easily combine that history with first-party cookies that it generates to identify a user in a unique way.

To assess the uniqueness of users, we generate for each user a cohortId per year between 1995 and 2019 using SimHash and SortingLSH with a number of bits ranging from 8 to 20. If the cohortId of a year is null due to the fact that the user did not rate any movie that year, we copy the cohortId of the

[6] https://www.chromium.org/Home/chromium-privacy/privacy-sandbox/floc

previous year. We evaluate the uniqueness of the cohortId path of users according to the number of bits of the hash using both SimHash and SortingLSH in Figure 7.



(a) SimHash

(b) SortingLSH

Figure 7. The uniqueness of the cohort path of users increases according to the size of this path, the larger the more unique.

The results (Figure 7 show a clear rise in the rate of unique users proportional to the size of the path of the cohortId whether it be in the case of SimHash or SortingLSH.

First, with SimHash the average unique user is more important when the number of bits used in the generation of the cohortId is bigger. This is explained by the fact that a larger number of bits allows for more differentiation and more possibilities in the outcome of the cohortId. Thus, two users with close but not similar interests (e.g., cars and motorbikes) are more likely to be assigned to different cohorts with a greater number of bits. On the other hand, the best result obtained during our simulations is around 23% of unique users. At first sight, this result might seem rather low but if we look more closely at the cohortIds generated for each year, we notice that a large part of the users have only one non-zero cohortId for the whole set of years over which the dataset is spread. Indeed, many users in the MovieLens dataset actually give few ratings between 1995 and 2019, as explained in the dataset section.

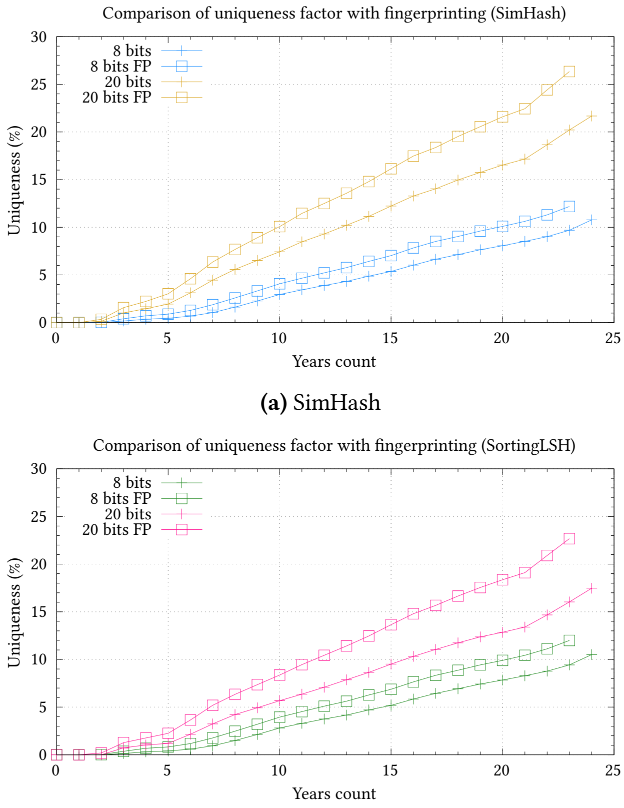
As for SortingLSH, we notice that the average number of unique users is slightly lower compared to SimHash. SortingLSH is implemented in such a way that each cohort respects k-anonymity (minimum number of individuals in a cohort), so it actually has very little impact on the uniqueness of a cohortId path.

When a user is browsing a web page, the website can retrieve more than its cohortId. In reality, each user has a browser fingerprint that allows a website to load correctly (depending on the browser). Much information can be part of the browser fingerprint and be used to uniquely identify a user [11, 15]. As described in the FLoC white paper, Google proposes to leverage the user-agent. The user-agent associated to a browser looks like:

```
  Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0)
Gecko/20100101 Firefox/95.0
```

The latter indicates to the visited site the name of the web browser used with its associated version but also the architecture, the OS name and version. To adapt our simulation to real world situations, we chose to assign each user a random user-agent containing only a web browser name. To choose the distribution of this random assignment, we based ourselves on the latest market share figures for web browsers.[7] The uniqueness using only the browser name as a fingerprint is depicted in Figure 8. Results show an improvement in the average uniqueness of a few percent for the SimHash and SortingLSH algorithms. Using more detailed browser fingerprints will improve the uniqueness of users.

[7] https://gs.statcounter.com/browser-market-share



(a) SimHash

(b) SortingLSH

Figure 8. Using the browser name as a fingerprint increases the uniqueness of users.

3.4 Discussions

This study is limited by the fact that FLoC is publicly documented through a research paper and a white paper, however it has not been publicly open-sourced. Our study focused on being as close to a real-world implementation as possible according to the white paper, and the source code of our experiments is publicly available [8]. Specifically, we settled on the dataset introduced in the Google white paper: the MovieLens 25M. After analyzing it, it turned out that in the context of calculating users cohortIds, the dataset provided enough data to generate consistent results in terms of cohort similarity in general. However, when we applied our simulations to study cohortIds' evolution, it turned out that the dataset was quite sparse.

Even if our simulations can give an indication of the performance and behavior of a system such as FLoC, they remain simulations that cannot be compared to a beta test period for example (e.g., origin trial) as point out by the observations of Criteo (Section 2.2). Moreover, the MovieLens datasets lack too many real-life elements to make the comparison relevant, for example, the browser fingerprint, which by itself could totally change the results of uniqueness as shown above. The difference of observations made during the origin trial and the ones obtained through simulation using the MovieLens dataset questions the validity of the results according to the considered data. This difference has certainly contributed to giving up the FloC project for replacing cookies for interest-based advertising by instead grouping users into groups of users with comparable interests.

In its place, Google today announced a new proposal: Topics [9]. The underlying idea of this new proposal relies on a computation on the browser of a handful of topics (e.g., fitness or painting) that represent the top interests of the user for that week based on the browsing history. Only topics of the last three weeks are kept, the older ones are deleted. Topics are selected only on the user device without involving any external servers. When the user visits a web site, Topics picks just three topics, one topic from each of the past three weeks, to share with the site and its advertising partners. The user also has control over the identified topics and can remove the unwanted ones. In addition, topics are curated to exclude sensitive categories, such as gender or race. Google expects to deploy and test this new proposal late in 2022.

Responding to the growing user demand for more privacy seems unavoidable. This certainly requires the deletion of third-party cookies which leads to user tracking on the Internet. However, the alternative solution adopted must be able to provide guarantees in terms of privacy. We have seen in our experiments that the cohort path can be used quite easily as a unique fingerprint to track users and therefore be an alternative to third party cookies (what FLoC is supposed to avoid). Moreover, the trade-off with the utility in a real condition does not seem to have enough value for an interest based advertising context. Finally, besides [8] explains that both SortingLSH and PrefixLSH could be decentralized, no clear protocols are proposed. FLoC relies on a centralized server (operated by Google) which requires access to all users' browsing history to partition and compute the cohortIds. This requirement in fact increases the knowledge of Google about our Internet browsing while it already captures a large part of our activities.

Other questions may also arise. Compared to cookies, which can be used by anyone, proprietary technologies such as FLoC or Topics raise questions of dependency for companies and dominance position for Google over the online advertising market. Besides, keeping personal data on the user's device and deporting part of the processing to the user's device (paradigm followed by many new federated learning schemes [2, 10]) are a clear step forward to an improvement of the privacy, we can ask ourselves the question of sharing topics of interest of users with advertisers when visiting a website (as proposed in Topics). Moreover, if companies prefer an alternative approach to the one proposed

[8] Anonymized

[9] https://github.com/patcg-individual-drafts/topics

by Google to target ads (e.g., to avoid too much dependency to Google), other intrusive and not transparent practices can be reinforced such as using fingerprinting to follow users.

4 Conclusions

The Movielens dataset used by Google for their experiments described in their white paper, despite containing a large number of records, remains rather inadequate for capturing the utility and privacy performances of FLoC in real condition (i.e., with user's browsing history which can be much more diverse and less sparse than the ratings of the MovieLens dataset).

About the utility of the system in terms of interest-based advertising, our results make us believe that it is very good, since the average cohort similarity is quite high compared to a random distribution and since users are quite stable in their cohorts. However, Criteo's observations based on the origin trail states something completely different.

As for our evaluation of the FLoC API in terms of privacy protection, it would seem that Google's newest technology presents some improvements compared to the third-party cookie system. However, it is far from being perfect since there are ways to use cohortIds to do cross-site tracking or personal inference about users. This cross-site tracking can be drastically improved by leveraging browser fingerprints (e.g. IP address, user-agent, etc.).

Although working on an alternative to third-party cookies is liable, the limitations in terms of utility and privacy of FLoC certainly weighed in on Google's decision to kill this project and to work on another interest-based advertising proposal.

References

[1] Andreas, K., Hugo, G., Tobias, B., Konrad, R., Felix, F.: Fingerprinting mobile devices using personalized configurations. In: PoPETS (2016)

[2] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, B., et al.: Towards federated learning at scale: System design. Proceedings of Machine Learning and Systems **1**, 374–388 (2019)

[3] Boutet, A., Ben Mokhtar, S.: Uniqueness assessment of human mobility on multi-sensor datasets. In: ARES (2021)

[4] Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing. p. 380–388. STOC '02, Association for Computing Machinery (2002)

[5] De Montjoye, Y.A., Hidalgo, C.A., Verleysen, M., Blondel, V.D.: Unique in the crowd: The privacy bounds of human mobility. Scientific reports **3**, 1376 (2013)

[6] Eckersley, P.: How unique is your web browser? In: PETS. pp. 1–18 (2010)

[7] Enev, M., Takakuwa, A., Koscher, K., Kohno, T.: Automobile driver fingerprinting. PoPETs **2016**(1), 34–50 (2016)

[8] Epasto, A., Muñoz Medina, A., Avery, S., Bai, Y., Busa-Fekete, R., Carey, C., Gao, Y., Guthrie, D., Ghosh, S., Ioannidis, J., Jiao, J., Lacki, J., Lee, J., Mauser, A., Milch, B., Mirrokni, V., Ravichandran, D., Shi, W., Spero, M., Sun, Y., Syed, U., Vassilvitskii, S., Wang, S.: Clustering for private interest-based advertising. In: KDD. p. 2802–2810 (2021)

[9] Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. **5**(4) (dec 2015)

[10] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021)

[11] Laperdrix, P., Bielova, N., Baudry, B., Avoine, G.: Browser fingerprinting: A survey. ACM Trans. Web **14**(2) (apr 2020)

[12] Manku, G.S., Jain, A., Das Sarma, A.: Detecting near-duplicates for web crawling. In: WWW. p. 141–150. Association for Computing Machinery (2007)

[13] Mathur, A., Acar, G., Friedman, M.J., Lucherini, E., Mayer, J., Chetty, M., Narayanan, A.: Dark patterns at scale: Findings from a crawl of 11k shopping websites. Proceedings of the ACM on Human-Computer Interaction **3**(CSCW), 1–32 (2019)

[14] de Montjoye, Y.A., Radaelli, L., Singh, V.K., Pentland, A.".: Unique in the shopping mall: On the reidentifiability of credit card metadata. Science **347**(6221), 536–539 (2015)

[15] Pugliese, G., Riess, C., Gassmann, F., Benenson, Z.: Long-term observation on browser fingerprinting: Users' trackability and perspective. Proc. Priv. Enhancing Technol. **2020**(2), 558–577 (2020)

[16] Rebekah, O., Rachel, G.: Blogs, twitter feeds, and reddit comments: Cross-domain authorship attribution. PoPETS pp. 155–171 (2016)

[17] Rescorla, E., Thomson, M.: Technical comments on floc privacy (2021)

[18] Research, G., ads: Evaluation of cohort algorithms for the floc api (2020), https://github.com/google/ads-privacy/blob/master/proposals/FLoC/FLOC-Whitepaper-Google.pdf

[19] Romanini, D., Lehmann, S., Kivelä, M.: Privacy and uniqueness of neighborhoods in social networks (2020)

[20] Rouzaud, A.: Floc origin trial: What we observed (2021), https://medium.com/criteo-engineering/floc-origin-trial-what-we-observed-3f7e8f209b82

[21] Zhao, K., Lu, H., Mei, J.: Locality preserving hashing. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. p. 2874–2880. AAAI'14, AAAI Press (2014)