A Privacy Analysis of Google's Topics Proposal

Martin Thomson
Mozilla
mt@mozilla.com

January 6, 2023

Abstract

The Topics API is a second attempt — after FLoC — to find a way to provide sites with information that can be used for ad targeting based on user interests. We find that the privacy protections in the Topics API are not sufficient to provide users with confidence that they cannot be reidentified and tracked. A critique of Google's initial privacy analysis is offered.

We also examine shortcomings of the topic witness mechanism. This mechanism is intended to protect privacy by limiting access to the information provided by the API. The mechanism produces both competition and privacy problems.

We conclude that Topics has significant and structural privacy challenges that are difficult to remediate.

1 Introduction

The introduction of Web features that enable the unconstrained cross-site flow of information on the Web, such as cookies, has been exploited for a number of purposes. The use of these mechanisms by advertisers and data brokers to assemble detailed records of the browsing history of large numbers of people has motivated changes to the platform to progressively remove these capabilities.

Though abrupt removal of tracking capabilities is possible, there is a prevailing desire to minimize the disruptive effect of a change on existing industries. In particular, online advertising has both benefited extensively from the cross-site flow of information and provided substantial financial support for Web content. Denying access to cross-site information for advertising purposes is generally regarded as having a significant impact on the profitability of all business that use, display, or support the provision of advertising.

Efficiency loss is a large part of why removing cross-site tracking has had an adverse effect on advertising. In advertising, efficiency is improved by:

- Measurement techniques provide feedback mechanisms that allow for managing expenditure and optimizing strategies.

- Targeting ensures that advertising is only shown to users who have either shown an interest in the topic or who are known to be receptive or susceptible to the chosen form of advertising.

These both use cross-site information. Finding substitute designs that support measurement is generally regarded as both more urgent and more tractable than addressing targeting uses. Without effective feedback on performance, advertising efficiency is greatly affected. Many advertising players assert that losing the ability to measure advertising has a significant effect on the viability of their business. Moreover, aggregated measurement techniques offer a way to provide measurement without enabling tracking of individual activity.

The use of targeted advertising is more controversial. Advertising generally use a bidding system to determine which advertisement is placed in a given context. Targeting algorithms adjust the bids for different advertisements based on several inputs:

Context — the page in which the advertisement will be showed, including the general topic area of the site or the specific topic for the page.

Demography — demographic information about the person viewing the page that can be obtained. Demographic data can be direct or inferred. Inferences use data from the site that will show the ad (first party data), but tracking might be used to link activity to other data sources or improve inferences.

Behavior — specific information about the past actions of this user. This includes first party data, but tracking can be used to access a history of activity on other sites.

The proposed Topics API [KK22] from Google Chrome aims to provide demographic information for use in targeted advertising. The API supplies sites with information about user interests. The amount of information that the API releases about interests is very limited, which is intended to provide some privacy for users.

2 Topics Overview

The Topics API provides information about topics that a user has shown interest in. The API infers interests based on the sites that a user visits. At the end of each week, the browsing history of users will be analyzed. All of the topics are ranked based on how often sites with the associated topic are visited. In the following week, sites can request a topic to randomly receive one of the top 5 ranked topics from each of the three preceding weeks.

A proposed taxonomy of 349 topics is used in the current proposal. Sites are allocated zero to three topics (usually one) based on the output of a machine learning model that uses a domain name as its sole input. This model is trained with data from a web crawler. Specific topics are provided for 10,000 popular sites.

The taxonomy and model Google has used in its trials of the API are proprietary, but Google have indicated an intent to open the processes used to construct a taxonomy

and assign it. This analysis makes no assumptions about the taxonomy or model, but uses the number of topics in Google's trial to calculate indicative values.

2.1 Privacy Measures

Privacy in the Topics proposal relies on both constraining and randomizing the information that is provided. The small set of topics limits the amount of information that sites receive as a result of learning any topic.

Drawing at random from the top 5 topics provides some amount of unpredictability. A random draw makes it more difficult for a site to gain a comprehensive view of the top 5 topics of any user.

The proposal also has a 5% chance of drawing at random from the full set of topics. This is intended to provide users with deniability.

The Topics API releases new information only once per *epoch*, which is set to one week in the proposal. Repeated calls to the API in the same week result in sites receiving the same result, providing no additional information.

Google describes controls that allow both users and sites to opt out. Users can ask their browser to disable the API. For sites, the permissions policy API [Cle20] can be used by a site to deny embedded cross-site content access to the API.

The final privacy measure is that topics are only revealed if a site was previously witness to the user visiting a site with that topic. A site that invokes the API does not receive any topics — aside from the purely random 5% draw — unless that same site invoked the Topics API on a site with that topic in the past. The stated reason for this measure is to ensure that the information provided by the Topics API is strictly less than the technology it is intended to replace, namely cookie-based tracking. We examine this mechanism in detail in Section 5.

2.2 Comparison to FLoC

Topics is Google's second attempt to propose an API that supports behavioral targeting, after FLoC [XK21]. Topics includes several design features that are in response to some of the feedback on FLoC:

- Providing comprehensible topics to sites is intended to make it easier for users and sites to understand the information that is being revealed. Feedback indicated that FLoC cohort identifiers were not easily understood. Using recognizable topics might also admit direct user control, either by promoting or hiding specific topics.
- The need to manage sensitive cohorts is something that curation of the set of topics is intended to address. The topic taxonomy is intended to avoid creating information about sensitive topics by not including sensitive subjects.
- FLoC was not strictly more private than cookie-based tracking in all cases. Restricting the topics that are revealed to those already witnessed by a site is intended to address this concern.

An accompanying analysis [EMIK22] is intended to inform decisions Topics by exploring how the API might interact with browser fingerprinting, as raised in a previous paper from Mozilla [RT21].

3 Response to Topics Privacy Analysis

Google has presented an initial analysis of the information release enabled by the Topics proposal [EMIK22]. This analysis looks at the aggregate release of information that results from sites using the API. The theoretical analysis is supported by numbers derived from browsing history of Google Chrome users. The data from Chrome is used to estimate aggregate information based on real browsing activity.

We provide some observations on this analysis that might require consideration when it comes to understanding the privacy properties of the Topics API proposal. One key observation is that the aggregates chosen for analysis do poorly in capturing the effect when smaller groups of people are selected. Given that there are many ways for smaller groups to occur, there appears to be no real limit on the worst-case privacy characteristics of the proposal.

3.1 Bounding Information Release

The main result in [EMIK22], Theorem 2, places an upper bound on the aggregate information release of $\log(N/k)$ or ≈ 6.13 bits[1] for each value. Observe however that this is an upper bound on the *aggregate* information release. This bound does not apply to the information released by a single invocation of the API, only an aggregate across many invocations.

It is trivial to construct an example a single use of the API releases more information than the aggregate bound.

Take the case where there are $|\mathcal{U}|$ users, with only one user ranking a topic t_2 in their top 5. We permit w_1 to learn $\overline{T_2}$, indicating that the site has a comprehensive view of the topics for all users on w_2. If just one item in $\overline{T_2}$ contains t_2, then the information gained by observing $t_1 = t_2$ on w_1 is $\log(|\mathcal{U}|)$. The information revealed about the single user exceeds the aggregate information release across any population when that population reaches just $\lceil N/k \rceil = 70$ users.

Noting that this portion of the analysis sets the probability of returning a random topic (p) to 0, we likewise disregard the effect of random topic selection. The efficacy of the random draw mechanism is discussed further in Section 3.4.

This somewhat contrived example assumes that user interests are relatively stable over time so that a site can learn $\overline{T_2}$. While not reliably true, this sort of assumption is necessary when considering the *worst-case* characteristics of a system. Stable values are hard to defend in a system that continuously releases information.

[1] Note that the paper says 6.12, but $\log_2(349/5) = 6.1251551313\ldots$

3.2 Limitations of Aggregate Statistics

Aggregate measures of information — those based on expectations or averages — produce *lower* values when a population is not uniformly distributed. A non-uniform allocation of identifiers to a population will produce less information in the aggregate than a uniform allocation. However, those users with rarer interests contribute more information ($I(x) = -\log(P(x))$) by revealing those interests. The information contribution of a single user is bounded only by $\log(|\mathcal{U}|)$, not any parameter choice.

Aggregate statistics like entropy[2] or KL-divergence[3] scale the information contribution of rare events by the (low) probability of those events. Users with rare choices suffer a larger loss of privacy when revealing their choice, but that contributes little toward these metrics. This makes these statistics deceptive for use in privacy analysis.

Aggregate statistics provide insight into the *utility* of the API. For instance, entropy represents what an advertiser learns relative to a starting point of total ignorance. A higher entropy value is likely to be more useful. Here, the data provided in [EMIK22] implies that the aggregate information gain over a very large population is likely small in practice, at around 1 bit for each iteration of the API, down from the maximum of 6.13 bits.

For the purposes of understanding the impact on privacy, low values for these aggregate statistics only indicate an uneven distribution of the information that is revealed. Little can be said about the effect on individuals.

To better understand the privacy impact of the API, the overall distribution of interests in topics across all users would provide more useful information than an aggregate statistics.

It might be tempting to also consider using distributions to inform the design of an API, including the choice of topics. For instance, a distribution might show which topics are least popular, which might be candidates for removal or merging with others. It might also identify over-represented topics, for which division into more specific topics might be advantageous for utility, which — for a popular topic — might be done without affecting privacy significantly. However, we will see in the next section how this sort of analysis might not be useful in informing these decisions.

3.3 Small, Biased Populations

The use of aggregate statistics can also hide the effect of a mechanism on small populations. The analysis in [EMIK22] samples browsing history from Chrome users. The low overall entropy of that dataset indicates that interests are not uniformly distributed even over a relatively large population.

Sites that use the Topics API are not necessarily presented with such a large and diverse population. Most sites will have smaller audiences that exhibit certain biases in their interests.

[2] $H(x) = -\sum P(x)\log(P(x))$

[3] $D_{\mathrm{KL}}(P \parallel Q) = -\sum P(x)\log\left(\frac{P(x)}{Q(x)}\right)$

With a sufficiently large population of users, some number of users will present as being interested in all topics. Users that are drawn from a very large population are likely able to reveal an interest in any topic and be confident that they are not revealing membership in a group that is too small to provide them acceptable privacy. As user populations become smaller or populations are selected for interests in certain topics, this confidence diminishes.

Sites that address a certain topic will generally attract visitors with an interest in that topic. This assumed tendency is the basis for how the Topics API proposes that browsers learn about user interests. However, this tendency means that visitors to a site will show a selection bias toward users with an interest in that topic. Interest in different subjects can be correlated, so sites that select for a topic will therefore reduce the prevalence of uncorrelated or inapposite topics.

A smaller population or a population with this selection bias might not have a sufficient number of users with uncorrelated interests to provide adequate privacy for a user that happens to reveal one of those interests. Users that reveal an interest in topics that are less well represented in the chosen population experience greater privacy loss as a result.

Even sites that have larger populations of users are able to select subsets of their population of visitors in order to improve their ability to create smaller groupings. This could use other properties to distinguish visitors, including IP geolocation, fingerprinting, or even the time of day that they visit.

Overall, this suggests that obtaining more precise data about how users might be distributed across topics — especially for large populations — does not result in actionable information about the privacy properties of the design.

3.4 Eliminating Noise

Producing a randomly selected topic with a probability of $p = 0.05$ effectively introduces differential privacy, albeit with the somewhat high $\varepsilon \approx 10.4$. This ε value effectively increases with multiple observations. If the interests of a user are stable, then multiple uses of the API might be used to eliminate the noise.

Repeat observations of the same topic allow sites to filter out noise with high confidence. A topic that appears n times for a user is the result of a random draw on all occasions with a $(p(N-1)/N)^n$ probability. Multiple appearances of a topic is therefore enough to increase confidence that a topic is genuine to $\approx 99.75\%$ after two repetitions or $\approx 99.988\%$ after three.

This is mitigated somewhat by the retention of topics in the API. Any topic that is revealed will be repeated for three weeks, each time being shuffled with topics from the other weeks. Apparent repetitions might then be the same topic from a previous week. Sites would need to observe the evolution of the set of topics over multiple weeks to correctly identify recurrent topics. Thus it takes repetition for 4 subsequent weeks or repetition over two non-contiguous weeks to recognize a true repetition of a topic.

3.5 Worst-Case Analysis

Example 1 from Section 4 of [EMIK22] claims to analyze a worst-case for privacy resulting from use of the Topics proposal. This example uses a contrived (and unlikely) example in an attempt to find an upper bound on the information that is revealed by the proposal.

In this example, topics are grouped into groups of $k = 5$, with the top k topics for each user being limited to those that are in the same partition. This does not appear to correctly identify a worst-case scenario by virtue of only considering aggregate information release.

Example 1 assumes a distribution of topic allocations that is $|U|/(N/k)$-anonymous, which does not capture a true worst case. Though it is reasonable to assume no prior information about the distribution of users to a site and thereby use a uniform distribution over $\mathcal{U}$, the same cannot be assumed for the distribution over the interests of those users.

We can construct a more contrived example to demonstrate a true worst case if we create a partition of topics $P' = S(U, k)$ where a single user shows interest, that is:

$$\left|\left\{S(u,k) : S(u,k) \cap P' \neq \emptyset\right\}\right| = 1$$

In the example as presented, the site knows that users are constrained in their choice of topics. Here, revealing that assumption and any topic, t_2, from that users interests also reveals that $\mathsf{i}_1(U) = \mathsf{i}_2(U)$. The maximum information release under this assumption is therefore bounded only by $|\mathcal{U}|$.

3.6 Learning About Unique Interests

The previous example assumes that sites are aware that a user has unique interests. It is informative to explore how sites might seek to learn that interests are unique or rare without prior knowledge. Without this knowledge the identity of the user might be protected unless sites become confident that the interest of that user in a revealed topic — or combination of topics — is genuine and unique.

As noted in Section 3.4, sites can gain high confidence that an interest is genuine in time. Gaining confidence that a user has a genuine interest in multiple topics takes more time.

Establishing that an interest in a single topic is unique is more challenging. A unique interest is one that no other user shares, so sites need to become confident that all other users have a genuine interest in k other topics. Gaining sufficient confidence that user interests are accurately reflected for the entirety of a population of any non-trivial size seems unlikely. Though we need to admit the possibility that some users have stable interests, we cannot rely on the interests of an entire population remaining stable over any time frame.

Concluding that a set of topics, $\mathcal{G} \subset \mathcal{C}$, is unique becomes easier as the set of genuine topics approaches k in size. If $|\mathcal{G}| = k$, then each user need only show a genuine interest in any topic outside of that set, that is, $t \notin \mathcal{G}$.

Though this demonstrates some challenges in recovering identity, the information a worst case reveals is not bounded. Under worst-case conditions, sites are able to establish a hypothesis about the correspondence of user identity between sites, using only the information that the Topics API reveals.

Once established, confirming a hypothesis about the correlation of identities requires very little information transfer. Furthermore, interests do not need to be unique for this information to be usable. Discerning between users in a small group does not require much more information than confirming a hypothesis about a single user.

3.7 Comprehensive Understanding of Interests

The number of possible combinations of top topics ($\binom{349}{5} \approx 2^{35.3} \approx 42$ billion) is greater than the number of Web users. For users with stable interests, it is possible that revealing all of their interests could be highly identifying.

The proposed design only hides the set of all interests on a probabilistic basis. Some unlucky users might reveal all of their topics to a site after three weeks, which requires just two interactions on non-adjacent weeks. For users who have a stable set of topics over the covered period, there is small $(1-p)^k \frac{k!}{k^k} \approx 2.97\%$ chance of them revealing all their top 5 topics.

The chance of revealing all topics increases with time and additional invocations of the API. Additional invocations also greatly increase the confidence that the observed topics are genuine (Section 3.4).

The resulting privacy loss might be mitigated by any drift in inferred interests for some users, which reduces the certainty a site might be able to attain about user interests. No data is currently available about how user interests, as measured by the proposed mechanism, might change over time.

3.8 Multiple Vantage Points

Two sites that have any means of linking user identity can share the topics they learn. This might more rapidly build a complete understanding of the interests of a user.

User identities that were linked across sites prior to the introduction of tracking protections can be used for this sort of information sharing. For new users, aside from unsactioned tracking techniques, the use of explicit interfaces that permit the joining of user identities across sites, such as federated login [Got22], are possible.

This suggests another shortcoming in the threat model adopted by [EMIK22]. The target of the analysis was to show that the correlation between linked identities across two sites was protected; that is, that it is hard to learn that $\mathsf{i}_1(U) = \mathsf{i}_2(U)$. This is a reasonable target, but it assumes that cross-site linkage does not exist. A more realistic goal, though a more challenging one, might allow for some number of sites to know pre-existing links between identities. That is, for two disjoint sets of sites, $\mathcal{A} \cap \mathcal{B} = \emptyset$, it is difficult to find any link between the per-site identity of a user for any site in set $\mathcal{A}$

and the per-site identity of the same user for any site in set $\mathcal{B}$:

$$\bigvee_{\forall a \in \mathcal{A}, \forall b \in \mathcal{B}} \mathsf{i}_a = \mathsf{i}_b$$

Establishing reasonable values for $|\mathcal{A}|$ and $|\mathcal{B}|$ here is challenging. Allowing larger sets are far more likely to gain a comprehensive view of users interests without noise, which produces worst case outcomes with much higher probability.

A favorable analysis for Topics might depend on having smaller sets. That means justifying the choice of set size. However, entities that currently employ tracking are known to have stores of data that allow them to link identities across large numbers of sites. Though the introduction of anti-tracking techniques might affect the viability of those stores over time, some amount of linking is likely to occur, making this hard to predict even in the long term.

Sites that are able to merge their views on topics might also counter the effect of drift in user interests as it allows for near-simultaneous observation of interest topics. This sort of information sharing would be effective in reducing the efficacy of the noise protections in the API.

4 Topic Assignment

The API relies on a centralized system for choosing and allocating topics. Here we examine the motivations and consequences of this design choice.

4.1 Topic Taxonomy and Sensitivity

Apple observed in their feedback[4], that the topics chosen for the trial exhibit certain cultural biases. Any system that relies on a taxonomy needs a system that can produce a set of topics free from these biases.

Any system that aims to eliminate this sort of bias creates other problems for the API. We show in Section 3.3 that the privacy risk that comes from revealing a topic increases as the likelihood of other users of a site having similar interests reduces. Any topic taxonomy that is representative of a diverse set of cultures — so that it can be useful within each of those cultures — will include topics that are underrepresented in communities where those cultures are absent or in a minority. This could greatly increase the amount of information revealed by the proposal for people from a minority.

Apple also point out that sensitivity to different topics can change based on a number of factors include culture, religion, age, community, and individual preferences.

We further observe that the sensitivity of a topic can be highly contextual. For example, user visiting a a site operated by an employer might find it distressing to have their interest in the "/Jobs & Education/Jobs/Job Listings" topic revealed. Consider also the case where a potential renter reveals their interest in sub-categories of "/Pets & Animals" to a housing agency with strict policies against pets.

[4] `https://www.mail-archive.com/webkit-dev@lists.webkit.org/msg30445.html`

4.2 Site Self-Selection of Topics

Sites are not permitted to present a list of topics that will be attributed to them. While this remains an open question about the API design, permitting sites to choose their own topics creates problems. In particular, it is difficult to ensure that sites present topics that are both *honest* and *consistent*.

Sites gain virtually nothing from learning that users are interested in the primary topic of their content. This is easily inferred from the fact that the user is visiting the site. The Topics API is valuable to a site only to the extent that it can reveal the topics a user was exposed to during visits to other sites that focus on different topics.

For any given site, there is no direct advantage to providing an honest choice for their own topic. However, presenting a false topic might result in users misrepresenting their interests to other sites, degrading the experience on those sites. Dishonesty thereby provides sites a small advantage by making ads on a dishonest site more relevant or less annoying by comparison to their competitors, who receive topics that are spoiled by the dishonest site.

In addition to spoiling the usefulness of the API for others, sites might offer different users a different set of topics. Inconsistent provision of topics might be exploited to use topics as a means of tracking by segregating populations of users using the API.

Segregating users by topic is a somewhat unreliable means of tracking as it depends on user activity across multiple sites. It is only useful to the extent that forcing a particular topic into the top 5 is possible.

4.3 Topic Boosting

A specially crafted subdomain can be used to allow a site to set a topic[5]. This depends on finding a subdomain for which the model produces the desired topic. The model ships in browsers and outputs are taken from a small space, so even a brute-force method would be sufficient to find a subdomain allows a site to select any topic it prefers.

One possible way to avoid having subdomains used for this purpose is to only provide a registrable domain[6] as an input to the model. This removes subdomains from consideration. The cost to usability is that this might reduce the effectiveness of the model in producing useful topics by providing less input. This also does not prevent attacks because it is relatively easy to create new registrable domains, by creating new names, by listing a higher-level domain on the public suffix list[7], or by creating names using services that are already listed.

Provided that users can be enticed to visit sites, controlling the domain allows sites to create false witness to topics.

More effort is required to use this technique to affect the ranking of topics, but this could be possible depending on how diverse the browsing history of each user is. Forcing a topic to a higher ranking than a topic that has been presented on numerous

[5] `https://github.com/patcg-individual-drafts/topics/issues/50#issuecomment-1135122153`

[6] `https://url.spec.whatwg.org/#host-registrable-domain`

[7] `https://publicsuffix.org/`

visited sites, the result of widespread browsing on sites that address that subject, is more difficult than displacing a topic that has only been presented on a few sites.

4.4 New Sites and Model Updates

The machine learning model that is used for allocating topics to sites is effectively a form of compression. Beyond the 10,000 sites that have explicitly assigned topics, sites will be allocated topics based on words or character patterns in their domain name.

Any model is likely to misclassify some names, especially for names outside of the training and validation datasets. If we assume that training is effective in teaching the model to recognize words that are associated with certain topics, then the model could be effective if those words are used. Performance for new names is unlikely to be better than a random allocation unless the name happens to include patterns or words that the model has been able to learn from.

Periodic retraining of the model is therefore necessary. This presents a continuous maintenance burden, particularly as it requires accurate labels for sites across all languages. If new sites for some languages appear more often than the labels and training occurs, users that visit those sites will present a less accurate representation of their interests. This could degrade the utility of the API for less popular languages.

Model updates will create some instability in the allocated topics for sites. This is unlikely to be a problem unless site names are chosen with the intent of generating a specific topic. This is unlikely to be successful for any purpose other than the attack described in Section 4.3.

5 Witness Requirements

As noted in Section 2.1, a major feature of the Topics proposal is that interests are only revealed to an entity that has previously witnessed an interest in that topic previously. This was specifically introduced to address concerns that the FLoC proposal was providing information that was not necessarily available to a site that uses cross-site cookies.

This witness restriction creates new problems for the proposal. We observe that this creates competition asymmetry (Section 5.1) and it also enables a new form of reidentification attack (Section 5.2).

These concerns only apply because of the condition that sites witness topics before they can receive them.

5.1 Topic Access for Sites

Several other commentators have noted the way that filtering out topics that a site has not seen in any context creates a natural bias toward entities that are able to be present on many page loads. It is proposed that only sites that have witnessed a user visit a site that has been allocated a given topic will be able learn that a user has an interest in that topic through the API.

Invoking the Topics API is necessary to register exposure to a topic. This requires that a cross-origin frame be created on the target site. Sites prefer to avoid creating frames to manage performance, which could reduce the number of entities might be allowed to act as witness.

For sites looking to gain access to information about interests across a wide range of topics there are a few strategies:

- Seek to gain a presence on as wide a set of sites as possible. The more sites the better, as presence on a site only counts if the site is able to witness a user visiting sites that are allocated each of their top 5 topics.
- Seek to purchase access to topics from another entity that has sufficient reach. This is technically trivial as it only requires instantiating a frame for the other entity, which can invoke the API and pass back the information.
- Seek to generate fake witness for all topics for each user. This might be possible using the techniques described in Section 4.3. While attempts to cover all 349 topics using redirects is likely to annoy users if done all at once, incrementally witnessing topics a few topics at a time might escape notice.

It appears then that there are several methods for circumventing these protections, but there is a natural advantage to sites that are already present on many sites.

5.2 Reidentification Using Topic Availability

In an issue on the proposal repository[8], Alexandre Gilotte observes that whether a site was witness to a topic carries information between contexts.

If a first site embeds content from a freshly created site, the fresh site can conditionally invoke the Topics API. In the following week, a second site can embed that same origin, which again invokes the Topics API. If the API returns the topic of the first site, then either this is a randomly-selected topic that happens to match that topic (probability 0.015%) or the API was previously invoked.

By creating multiple fresh sites, multiple bits of information can be conveyed using this technique. In the first week, the fresh sites either invoke the API or not in a pattern that carries information. In the second week, the same sites read that data back on the second site, transferring that information from the first site to the second.

This attack is only conditionally effective. The topic of the first site needs to be in the top 5 to even be considered. Then the topic needs to be selected from the top 5 (a 19% chance after the random draw is considered). A number of techniques are explored on the issue to push selected topics into the top 5. With these techniques, if the information is used to carry user identities, then $\lceil \log_2(|\mathcal{U}| + 2) \rceil$ bits is sufficient to identify all users. Around 5% of users will avoid this identification by providing a random topic.

Finally, reading information in this way makes the fresh site unusable for future iterations of the attack, at least as far as the shared topic of the two sites is concerned.

[8] `https://github.com/patcg-individual-drafts/topics/issues/74`

Attempting to read the value on the second site establishes that site as a witness for the topic of that site, even if it was not previously. Each attempt requires that new sites be created[9].

6 Potential Alternatives

Google's FLEDGE proposal [Kle22] describes a system whereby targeting information is not released to sites. The problems arising from releasing information about interests might be partly addressed by only making that information available to sites in an isolated context.

The "fenced frames" in FLEDGE create an isolated context that cannot release information, except under strictly controlled conditions. Within this context, sites can access private information for use in selecting ads. The display of the chosen ads is similarly isolated. Limited reporting channels allow for measuring reach, frequency, conversion, and other important metrics, while maintaining a determined level of privacy.

A similar mechanism might address some of the privacy problems of Topics. Isolating this information could contain the information released by users. This might prevent reidentification (Section 5.2) and disproportionate information release by users with rare interests (Section 3.3). It would not, however, avoid the attacks described in Section 4.3.

Isolating access to information might allow for richer taxonomies (Section 4.1) with less need to limit access based on an imperfect understanding of sensitivity.

The cost of isolation is increased complexity for implementation and use. In particular, FLEDGE requires that sites present only a few ads to the isolated context. This limits what can leak out of isolation through reporting channels. Information about user interests might be needed to guide the choice of these ads, which might be unmanageable.

7 Conclusions

The Topics API proposal provides sites with information that is derived from a user's browsing activity. Several mechanisms in the API are designed to ensure that the information is not identifiable.

In Section 3 we explore how Google's analysis of the information revealed by the proposal [EMIK22] concentrates only on the aggregate effect of the API. No consideration is given to the potential for disproportionate impact on users with interests that are less-well represented in any given population.

The random elements of Topics appear to make it hard to directly exploit the information that is provided for tracking, but any claims in this regard depend on user interests changing before a comprehensive view is obtained (Section 3.7). Users with stable interests are identifiable given sufficient time.

[9] This is not especially difficult: any number of new sites can be created under a domain that is registered on the public suffix list.

The information released by Topics is small in the aggregate, something that is supported with data in [EMIK22]. However, this suggests a potential for poor utility for advertising more than it suggests privacy advantages.

Methods are found whereby sites might alter both their own topic assignment, or even the top topics a user presents in Section 4.3.

The restriction on access to topics based on previous witness of the topic has been found to have serious privacy consequences (Section 5.2). The restriction also creates a bias toward actors with presence on more sites, which could favor large, incumbent entities (Section 5.1). We see no option other than to remove this mechanism from the proposal, even though that would recreate a privacy problem identified with FLoC that Topics set out to address.

Overall, there is no evidence to support any claim that the effect on individual privacy is acceptable. There is some basis for the opposite conclusion: we identify challenges that might render system-level analysis of any privacy claims infeasible. These challenges appear to be fundamental. It might not be possible to make small adjustments to the underlying design to prevent the sorts of leaks that have been identified.

References

[Cle20] Ian Clelland. Permissions Policy. `https://w3c.github.io/webappsec-permissions-policy/`, Dec 2020.

[EMIK22] Alessandro Epasto, Andres Muñoz Medina, Christina Ilvento, and Josh Karlin. Measures of cross-site re-identification risk: An analysis of the Topics API Proposal. `https://raw.githubusercontent.com/patcg-individual-drafts/topics/main/topics_analysis.pdf`, Mar 2022.

[Got22] Sam Goto. Federated Credential Management API. `https://fedidcg.github.io/FedCM/`, Jul 2022. Draft Federated Identity Community Group Report.

[KK22] Josh Karlin and Michael Kleber. The Topics API. `https://github.com/patcg-individual-drafts/topics`, May 2022.

[Kle22] Michael Kleber. First Experiment (FLEDGE). `https://github.com/WICG/turtledove/blob/main/FLEDGE.md`, May 2022.

[RT21] Eric Rescorla and Martin Thomson. Technical Comments on FLoC Privacy. `https://mozilla.github.io/ppa-docs/floc_report.pdf`, June 2021.

[XK21] Yao Xiao and Josh Karlin. Federated Learning of Cohorts. `https://wicg.github.io/floc/`, Apr 2021. Draft Web Incubation Community Group Report.