Let’s Talk About FLoC

PEARG – IETF 111
Josh Karlin @ Google Chrome

The web is partitioning by top-frame site

- Caches
- Socket pools
- TLS session resumption identifiers
- Cookies
- Javascript Storage
- etc..

- What happens on one site, stays on that site

Supporting Important Use Cases

- Studies have shown that without third-party storage, sites lose ~50% of their programmatic display advertising revenue
  - This is due to the degradation of interest based advertising, retargeting, and frequency capping capabilities. Not to mention loss of conversion reporting or spam and fraud prevention.
- There are lots of non-advertising use cases for third-party widgets as well:
  - Third-party login, payment providers, media, docs, etc.
- Chrome must both build the walls, and support these use cases to support the open web so that it can remain accessible and open.
  - We support these use cases with proposals for FLoC, FLEDGE, Conversion Reporting, Aggregate Reporting, Fenced Frames, Privacy Budget, IP masquerading, First Party Sets, Shared Storage, Trust Tokens, etc..

The problem FLoC is solving

- FLoC is interested in recovering interest-based advertising
    - In an easy-to-use way
    - Such that individual users remain hard to track

How it works today

- Today ad-tech runs script on the pages you visit
- The backend takes contextual signals about those visited pages, and joins them together via your user identifier (third-party cookie) which is the same across sites
- This data gets fed to a model which produces a prediction of what ad is likely to work best for you

How it could work with FLoC

- We can preserve the basic model, and improve privacy
  - Instead of ad-tech analyzing individual users
  - Reason about groups of users that the browser deems similar
  - And the group (cohort) is determined by the client (browser)
- Advantages
  - Much improved user privacy (groups of thousands of users)
  - Doesn’t require ad-tech backends to be rewritten
  - The server no longer knows *your* browsing history

The API

document.interestCohort()

- Returns a dictionary with a user cohort and version string
    - Version used to describe the algorithm used and any floc field trial value
- Rejects if:
    - User has no calculated cohort
    - Browser determines user has a sensitive cohort (this isn’t seen by Google)
    - User is in incognito
    - Document does not have interest-cohort permission policy
    - User has disabled privacy sandbox APIs

Deriving a Cohort

- **Goals**
  - Convert a list of domains of sites the user visited into one of ~32k clusters (cohorts).
  - Entirely client side.
  - Each cohort should have thousands of users.
  - Each cohort should not reveal sensitive information.
  - Each cohort should not provide significant fingerprinting surface.

The clustering algorithm

1. Step 1: Encode the user’s history
   - Convert each domain into a point in 64 bit space by hashing it
   - Create a sparse vector of 1s for domains user visited, 0 elsewhere
2. Step 2: Reduce to 50 bits via simhash
   - Create 50 random vectors of 2^64 dimension
   - For each vector, dot product it by the user’s history, and take the sign of the result as your output
   - The output is the ith bit
3. Step 3: Reduce to ~16 bits via prefix lsh
   - Google server distributes a mapping of simhashes to final 16 bit cohorts
   - The mapping ensures at least k users per cohort
   - The mapping removes cohorts that aren’t t-close to the general population for any sensitive category

Which pages are eligible for cohort calculation?

- Pages that use the API will be opted in
  - If they have public IP addresses and the user hasn’t opted out
  - If they aren’t opted out via the permissions API header
- During the Origin Trial, we included sites that had ad resources on them
  - As determined by Chrome’s Ad Tagging service
  - So that the OT would be representative of a launched API for those testing it

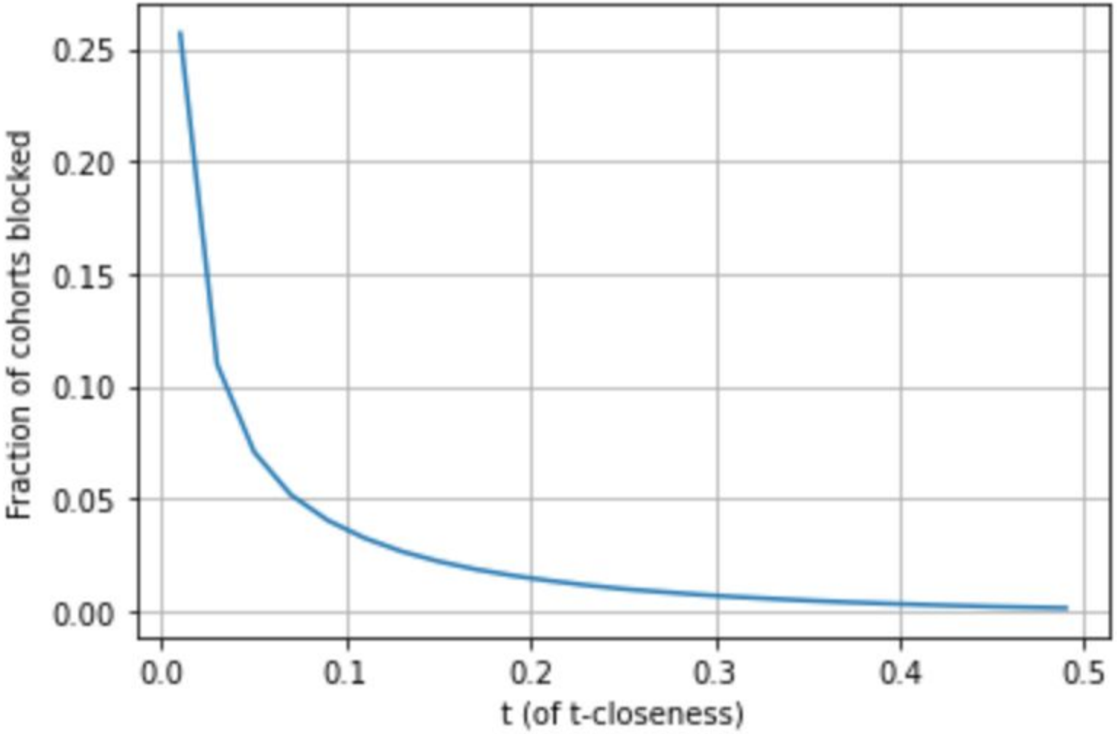
Ensuring k-anonymity of individual cohorts

- Ensured at least 2,000 chrome sync users per cohort
    - Using Chrome Sync data
    - Once we have aggregate reporting, could use that instead

Prevent leakage of sensitive categories

- We can capture many sensitivities
    a. By looking for correlations between cohorts and greater than normal browsing of with sites of a given sensitivity
    b. This can be done anonymously with sync data
    c. If a cohort isn't t-close to the general population for all sensitivities, revoke the cohort
    d. Revoked cohorts are distributed to clients
        - Clients then determine if their user's cohort is sensitive

Sensitivity analysis



What did the Origin Trial look like?

- Algorithm: One-hot simhash w/ prefix sorting LSH
- Cohort calculated once every 7 days
- Cohort includes 7 days of history
- Cohort is global across sites
- Clearing cookies or history clears cohort
- Cohort must have at least 2,000 sync users (so more in full population)
- Cohort calculation must include at least 7 different sites
- Pages with ads and those that use the API are included in calculation
- Pages can opt out via permissions policy
- Users can opt out via privacy sandbox setting

Feedback Received (Room for Improvement)

- Don’t auto-opt-in sites with ads in experiments
- Cohorts are hard to understand for end-users and technologists
- FLoC cohorts represent fingerprintable surface, can it be reduced?
- Can we further reduce the possibility of revealing sensitive information?

Possible Mitigations

- **Don’t auto-opt-in sites with ads in experiments**
- Done.

Possible Mitigations

- **Cohorts are hard to understand for end-users and technologists**
- Considering providing topics based on domains instead of cohorts
    - e.g., “/Arts & Entertainment/Performing Arts” or “/Beauty & Fitness/Fitness” as opposed to cohort 21849
    - Topics taxonomy could be curated (better for sensitivity)
    - Topics taxonomy could be much shorter (say ~256)
    - Topics are understandable, and the granularity is understood
    - Perhaps users could opt into or out of particular topics
- We’re seeing others talk about topics as well
    - e.g., Ad Topic Hints as proposed in the Privacy CG

Possible Mitigations

- **FLoC cohorts represent fingerprintable surface, can it be reduced?**
- If we went with topics, then a sample might represent ~8 bits instead of ~16
- Could provide a random FLoC w/ small probability (e.g., 5%)
- We can give different topics to different sites
  - e.g., provide a random one of the user's top-5 topics to a site for an epoch
  - Then 80% chance that two sites will have different topics for the same user
- Regardless of FLoC, fingerprinting is real and we're seeing it happen. That needs to be addressed
  - Please see Chrome's privacy budget proposal.

Possible Mitigations

- **Can we further reduce the possibility of revealing sensitive information?**
- Topics would be human curated to ensure they’re generally not sensitive
- Taxonomy would ideally be created and maintained externally in the long run
- We’d still likely want to perform server-side analysis to ensure that topics are t-close

Further room for improvement

Can we reduce the scope of the topics?

- So far FLoC has been a global value, derived from all sites that use the API
- Today, interests are derived via third-party cookies, based on a single third-party's view of the user's browsing
- We could reduce the scope of FLoC to be per-third-party, making it a strict subset of the capabilities of today's third-party-cookies
  - e.g., ad-tech A gets a topic associated only with the sites ad-tech A used FLoC on
  - Would need to limit number of third parties that could get FLoC per site per epoch too prevent too much fingerprinting data from being revealed
- The trade-off being that you get more samples per site, so we'd have to divide the epochs-to-fingerprinting by ~3

Thank you

For more information please see:

- Technical discussion and issue tracker: https://github.com/WICG/floc
- Sandbox Overview: https://privacysandbox.com
- FLoC Details and analysis: https://www.chromium.org/Home/chromium-privacy/privacy-sandbox/floc