Facebook Privacy-Protected URLs light Table Release

Solomon Messing Christina DeGregorio Bennett Hillenbrand Gary King
Saurav Mahanti Chaya Nayak Nathaniel Persily Bogdan State
Arjun Wilkins

September 2019

This document describes the **Facebook Privacy-Protected URLs-light release**, resulting from a collaboration between Facebook and Social Science One. It was originally prepared for Social Science One grantees and describes the dataset's scope, structure, and fields.

Citation

Messing, Solomon; DeGregorio, Christina; Hillenbrand, Bennett; King, Gary; Mahanti, Saurav; Nayak, Chaya; Persily, Nathaniel; State, Bogdan; Wilkins, Arjun, 2019, "Facebook Privacy-Protected URLs light Table Release", https://doi.org/10.7910/DVN/ZT0WZY, Harvard Dataverse, V3

Status

The data and data infrastructure that will be used to provide query access are ready for testing and initial use by external researchers.

Data Access

To obtain access to these data, see the Request for Proposals process at `https://socialscience.one/rfps`. No other means of access is currently allowed.

Summary

The data describe web page addresses (URLs) that have been shared on Facebook starting January 1, 2017 up to and including February 19, 2019. URLs are included if shared with public privacy settings more than on average 100 times ($\pm Laplace(5)$ noise to minimize information leakage). We have conducted post-processing on the URLs (as detailed down below) to remove potentially private and/or sensitive data.

The unit of analysis for these data is the URL. All aggregates are taken over the entire time period as described above.

The data set is about 7 gigabytes, comprising approximately 32 million URLs, and about 544 million cell values.

Data from users who have chosen to delete their accounts are not available due to legal constraints (and availability). Missing data may therefore have a larger impact on URLs that were shared further in the past. This dataset includes content that has been taken down due to Community Standards violations. We have developed standard operating procedures for preservation and replication of researcher results that anticipates these issues.

Infrastructure and Resources. Facebook has provided researchers with accepted proposals access to a system that provides an interface to query and analyze these data.

Recommended capabilities. Research teams should have experience working with data sets that do not fit into memory. Specifically, teams will need to understand what kinds of queries and analyses are expensive and likely to exhaust the resources of our computing cluster. They will also need to write R and/or Python analysis code that does not exhaust system RAM (e.g., on a modern server with around 64GB RAM). Having at least one individual with experience using SQL/HQL, Python, and Linux is highly recommended.

Privacy Protection

We are releasing aggregates in these data using a privacy preserving technology called *differential privacy*. Roughly speaking, data aggregates are differentially private if no specific person or action stands out in the data—if including or excluding a single user (action) does not change the result beyond a formally-specified point. Differential privacy thus provides plausible deniability and stymies re-identification attacks.

These guarantees are generally operationalized by adding noise to aggregates in such a way that provide mathematical guarantees about how well an individual or action is obscured. A non-technical introduction to differential privacy is available here: http://privacytools.seas.harvard.edu/files/privacytools/files/pedagogical-document-dp_0.pdf; a rigorous introduction can be found here: https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf.

Action-level zero-Concentrated Differential Privacy (zCDP). Data aggregates that describe user actions are protected under a form of action-level, zero-concentrated differential privacy (zCDP, see Bun and Steinke [2016]).

This action-level protection differs from the standard ("user-level") differential privacy guarantee in that the granularity of what is protected is not a single user, but rather a user single action (e.g. sharing a single URL).

Under zCDP, the parameter summarizing the privacy guarantee is ρ, which is achieved using the Gaussian mechanism—in other words adding $N(\xi, \sigma)$ noise to the data. Here the ξ term is zero and we set $\sigma = 200$, yielding an action-level $\rho = 1.25 \times 10^{-5}$.

We can translate this to the more familiar epsilon-delta differential privacy framework for ease of interpretation. We use the following (Lemma 3.5, Bun and Steinke [2016][1]):

$$\epsilon = \rho + \sqrt{4\rho \log(1/\delta)}$$

where

$$\rho = \frac{1}{2\sigma^2}$$

[1] Again, the ξ term is zero so we do not include it.

At the action-level, this translates to $\epsilon = 0.02$ using $\delta = 10^{-4}$.

Relation to user-level differential privacy. Users who have taken at most k actions summarized in variables describing actions surrounding URLs below will be protected under a differential privacy guarantee similar to user-level privacy. This relies on the fact that a user-URL-action can occur only once, due to the way we have constructed the data—for example, a user can share a single newspaper article URL only one time.

We can map this to user-level privacy as follows. The l_2 sensitivity for any user who has taken at most k actions is $\sqrt{k}$. Thus adding Gaussian noise with σ in each cell in these aggregated data satisfies $(k^2\rho)-zCDP$ (see Proposition 1.9, Bun and Steinke [2016]):

$$\rho = \frac{k}{2\sigma^2}$$

And we can plug this into the equation above (Lemma 3.5, Bun and Steinke 2016) to provide a user-level privacy equivalent ϵ:

$$\epsilon = \frac{k}{2\sigma^2} + \frac{\sqrt{2k\log(1/\delta)}}{\sigma}$$

For $\sigma = 200$ and $k = 500$, this user-level conversion translates to $\rho = 0.00625$.

Under these assumptions, setting $\sigma = 200$ and $\delta = 10^{-5}$ yields $\epsilon = 0.5$.[2] Users who have taken at most $k = 500$ actions summarized in variables describing user actions below will have formal privacy guarantees at $\epsilon = 0.5$. For the URL shares variable, this applies to more than 98% of users in the data set.

If a user actually made $k' > k$ actions, their ρ will be larger by a factor of k'/k. And to get their effective ϵ, we replace k' with k in the formula above. For example, if for a given person, $k' = 1000$, this person's effective ϵ will not be 0.49, but instead will be 0.69. If $k' = 2000$, effective ϵ will be 0.98, etc (still assuming weve fixed δ at inverse 10 thousand).

Implementation. We operationalize this protection by adding Gaussian noise to the individual-level aggregations (counts) in the data of the form $N(0, \sigma = 200)$. The distribution of noise added is summarized in the histogram below:

Of course, some values in DP-protected aggregated fields will be *negative* due to the noise added. While truncating these counts at zero will not affect privacy, doing so will bias statistical estimates.

We generate this noise using the Yarrow-160 cryptographically secure pseudo-random number generator [Kelsey et al., 1999]. Our implementation relies on os.urandom(), which reads noise from the /dev/urandom device in the Linux kernel, blocking noise generation until the system collects 128 bits of entropy from the urandom entropy pool. The idea is to gather "environmental noise," including inter-keyboard timings, inter-interrupt timings, and other non-deterministic events that are difficult for an adversary to measure or observe [Torvalds, 2014]. The device gathers randomness from these sources and adds them to an entropy pool, which it mixes using a function similar to a cyclic redundancy check.

We use the Marsaglia polar transformation method to generate Gaussian noise [Marsaglia and Bray, 1964]. The method works by generating two uniform random variables (x, y) over the square $1 < x < 1, 1 < y < 1$,

[2] setting δ to 1/1000 yields $\epsilon = 0.42$. Or, we can set δ to something much smaller, say one-millionth, which given the noise added, $\sigma = 200$, translates to $\epsilon = 0.59$.

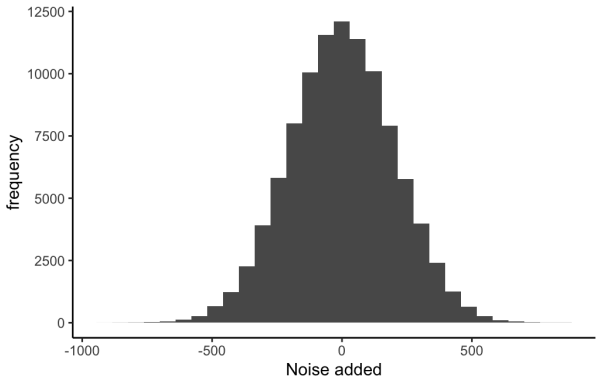

Figure 1: Gaussian noise added

until both points fall within the unit circle such that $0 < s = x^2 + y^2 < 1$. Then $x/\sqrt{s}$ and $y/\sqrt{s}$ correspond to the cosine and sine of the angle formed by (x, y). We then project $x/\sqrt{s}$ to polar coordinates using $\sqrt{-2ln(s)}$, and further multiply by the standard deviation σ and add the mean μ, which is zero here.

It should be noted that even without applying differential privacy, the sensitivity of these data are limited due to the fact that access is restricted to grantees and provided in a secure environment; that all URLs included have been shared at least 100 times $\pm Laplace(5)$ noise with fully public privacy settings; and that we've taken steps to remove unintentional PII from URLs and ensure they contain only navigation-critical information as outlined below.

Data

Fields to be included in data release. URLs will be included in the data if they have been shared publicly—meaning a user chose to share a URL in a post accessible to everyone on Facebook—more than on average 100 times (+ Laplacian noise with scale = 5 is added to minimize information that an attacker could use by exploiting a hard cutoff).

Fields with differentially private noise are followed by the abbreviation "DP." Note that some values in DP-protected aggregated fields will be *negative* due to the noise added. Note that truncating these counts at zero will bias statistical estimates.

An example dataset can be found at https://docs.google.com/spreadsheets/d/1RgfSTOe__kLAy3HlUY_BVAGHrYtHeEl197Xz_GcphKk/edit?usp=sharing; it may be helpful in understanding the fields described below.

- **url_rid [text]**: a unique URL id created specifically for this data set.
- **clean_url [text]**: The webpage URL after processing. This is the full URL (e.g., https://www.nytimes.com/2018/07/09/world/asia/thailand-cave-rescue-live-updates.html), not just the domain. URLs that are no longer reachable will persist in the data. The URLs have been processed in an attempt to consolidate different web addresses that point to the same URL and to remove potentially private and/or sensitive data. The following post-processing has been applied:
  1. Redirects are followed to the terminal URL, including URL shorteners.
  2. If the terminal webpage has an "og:url" meta-tag, the associated URL becomes the consolidated URL—often referred to as the "canonical URL." If not, the rel = "canonical" tag is used. If neither tag is provided, the canonical URL is taken from the raw URL address.

3. The vast majority of obvious PII (personally identifiable information) contained in URLs is already removed by virtue of filtering URLs to those with on average 100 public shares, since less frequently shared URLs contain the bulk of PII.
4. For urls with query strings (about 22% of URLs above), special processing is applied. A query string in a URL passes data to the server when a client requests content, for example the "v=Ipi40cb_RsI" in https://www.youtube.com/watch?v=Ipi40cb_RsI. Sometimes query parameters provide navigation data, which tells the server what content to deliver to the client, as above. However, query parameters can also pass to the server data irrelevant to navigation, such as whether a URL was accessed from Twitter or Reddit, tracking data, and/or PII. We have attempted to remove query parameters unrelated to content navigation by iteratively removing each query parameter and testing the resulting content for differences with original page content (above and beyond the difference introduced by re-loading the page, which can occur due to ads, 'suggested content,' and/or randomized menu options). Note that for the vast majority of URLs, removing these parameters does not result in content that is different from the original. This is done at the domain level for 100 URLs (unless the domain has fewer than 100 URLs in the data).
5. We keep query params that result in a different page title AND html content that differs by more than 2%, OR content that is > 95% different from original page. This measure is based on the difflib Python library and is defined as 2.0 * M/T, where M is the number of sequence matches and T is the number of elements in both sequences.
6. All URLs from domains that consistently fail to return a valid response within 120 seconds or consistently return a response under 100 characters are stripped of all query parameters.
7. Query parameter values that contain common phonenumber patterns are removed using the phonenumbers Python library.
8. We use regular expressions to remove email addresses that appear in any part of the URL string.

Example URLs. Left raw, right processed. Non-essential query values have been altered to protect privacy.

Raw URL	Processed URL
https://media1.tenor.co/images/da7eb8198618472aa82151e5d704f521/tenor.gif?itemid=5265827	https://media1.tenor.co/images/da7eb8198618472aa82151e5d704f521/tenor.gif
https://www.pivot.one/app/invite_login?inviteCode=csdfeddshkuyfckyc	https://www.pivot.one/app/invite_login
https://www.youtube.com/watch?v=oXWsoqesw7A&feature=youtu.be	https://www.youtube.com/watch?v=oXWsoqesw7A
https://www.youtube.com/watch?v=oX_fLP191-k&list=RDoX_fLP191-k	https://www.youtube.com/watch?v=oX_fLP191-k
https://news.google.com/newspapers?nid=2478&dat=10260530&id=xFc1AAAAIBAJ&sjid=iiUMAAdFJSIBAJ&pg=1558%2C27085012&hl=en	https://news.google.com/newspapers?id=xFc1AAAAIBAJ&pg=1558%2C27085012

- **parent_domain [text]:** parent domain name from the URL (eg. foxnews.com).
- **full_domain [text]:** full domain name from the URL (eg. www.foxnews.com, video.foxnews.com, nation.foxnews.com, insider.foxnews.com).
- **first_post_time [timestamp]** - The date/time when URL was first posted by a user on Facebook. Date-times will be truncated to 10 minute increments. The exact format is YYYY-MM-DD HH:MM:SS, for example: 2015-12-02 18:10:00.

- **first_post_time_unix [unix timestamp]** - The above field translated into unix time---the number of seconds since 1970-01-01 00:00:00, for example: 1449079800.
- **share_title [text]:** Provided by the author of the URL's content pulled from **og:title** field in original html if possible).
- **share_main_blurb [text]:** Provided by the author of the URL's content (pulled from **og:description** field in original html if possible).
- **tpfc_rating [text]:** If URL was sent to third-party fact-checkers, did they rate it (NULL if not) and if so, how did they rate it? Category values include: 'True', 'False', 'Prank Generator' , 'False Headline or Mixture', 'Opinion', 'Satire', 'Not Eligible, 'Not Rated.' Definitions, and a list of fact checkers, are available here: https://www.facebook.com/help/publisher/182222309230722 and https://www.facebook.com/help/572838089565953?helpref=faq_content. More information on how news that may be false is selected can be found here: https://www.facebook.com/help/1952307158131536. Only available for some stories, only available in Argentina, Brazil, Cameroon, Canada, Colombia, Denmark, France, Germany, India, Indonesia, Ireland, Italy, Kenya, Mexico, Middle East and North Africa, Netherlands, Nigeria, Norway, Pakistan, Philippines, Senegal, South Africa, Sweden, Turkey, UK, US. When more than one rating is given to a story, we use Facebook's *precedence rules*, described below.
- **tpfc_first_fact_check [timestamp]:** the date-time that article was first fact-checked, if at all. If the article has not been fact checked, this will be NULL. Date-times will be truncated to 10 minute increments. The exact format is YYYY-MM-DD HH:MM:SS, for example: 2015-12-02 18:10:00.
- **tpfc_first_fact_check [unix timestamp]** - The above field translated into unix time---the number of seconds since 1970-01-01 00:00:00, for example: 1449079800.
- **total_public_shares [integer] DP**: total number of unique accounts that shared the URL with public privacy settings.
- **total_spam_usr_feedback DP**** **[integer]**: the total number of unique users who reported posts containing the URL as spam.*
- **total_false_news_usr_feedback DP**** **[integer]**: the total number of unique users who reported posts containing the URL as false news.*
- **total_hate_speech_usr_feedback DP**** **[integer]**: the total number of unique users who reported posts containing the URL as hate speech.*
- **prop_share_without_clicks [double] DP**: number of users who shared a URL in a post or re-share, but for which there is no record of clicking on the link, divided by the total number of unique users who shared the post containing the URL. It's important to note that users often find URLs from a source outside of the platform. Nonetheless, this number may help identify articles that users share without reading, or URLs used in organized campaigns to spread content.
- **public_shares_top_country*** **[text]**: URL shares are tallied by country and the country with the most shares is provided as an ISO 3166-1 alpha-2 letter code. This field is *not* indicative of all locations where this article was posted—rather it is only the *top* country among users who shared it.

Third Party Fact-Checker Ratings and Precedence Rules Explained:

Based on a single fact-check, Facebook can reduce the distribution of a specific piece of false content. Facebook also uses similarity detection methods to identify duplicates of debunked stories and reduce their distribution as well. Facebook can use this as a signal to reduce the overall distribution of Pages and web sites that repeatedly share things found to be false by fact-checkers. Facebook is able to get useful signals

about false content that we can then feed back into its machine learning model, helping it more effectively detect potentially false items in the future.

Occasionally, multiple fact-checkers apply different ratings to the same piece of content. In these cases, the more definitive rating takes precedence, e.g. 'False' or 'True' trumps 'Mixture'. In very rare cases where the two most definitive ratings, 'True' and 'False', are applied to the same piece of content, 'True' takes precedence since we refrain from demoting content rated 'True' by a fact-checking partner. Our tpfc_rating incorporates the below precedence rules when deciding how to handle multiple fact checker ratings for the same URL. It is very rare for multiple fact-checkers to rate the same URL.

For third-party fact-checked content, a fact-checker in a country other than the top public shares country may have rated content if it circulated broadly within their country. For a complete list of our third party fact checkers, please visit this website and this one.

Instance	Example	Rule
Same Fact Checker, Multiple Ratings	A publisher appeals to the fact checker or the publisher updates the content, causing the fact checker to change its rating of the content	Use the rating with the latest timestamp
Many Fact Checkers, one rating per fact checker	Multiple partners fact check the same claim	Use the rating that wins the following precedence rule: True > False or Prank Generator > False Headline or Mixture > Not Eligible or Satire or Opinion > Not Rated
Many Fact Checkers, more than one rating per fact checker	Multiple partners have fact checked the same claim and some or all have revised their initial rating of the content	First take latest rating for each Fact Checker, then decide according to the same precedence rule as above using the latest ratings only: True > False or Prank Generator > False Headline or Mixture > Not Eligible or Satire or Opinion > Not Rated

**Note about user feedback fields: these fields constitute information provided by users, which may not be indicative of actual violations of Facebook's Community Standards, and like any survey question or coding exercise, may not be a measure of the concept the researcher intends. For example, for the variable "total_hate_speech_usr_feedback", users may share URLs to endorse or oppose the content. Endorsements of hate speech violate Facebook's community standards policy, while opposing it does not. Users may also flag content as hate speech because they disagree with it (if they perceive the difference or can distinguish if they do), rather than to actually indicate hate speech, resulting in ambiguous or false positive reports of hate speech, if taken literally. Similar issues apply to other fields.

For "total_spam_usr_feedback", in contrast to URLs found to contain hate speech (which Facebook deliberately does not block due to the subtleties above), URLs containing content that violates spam policies are blocked from the platform for future sharing.

These data were originally collected or derived from operational information or data sources or otherwise — and not for research purposes. Features of the dataset may be inaccurate, incomplete, or have been collected in ways that are not compatible with research goals. Researchers need to adapt their methods, research designs, and quantities of interest to the data at hand. Please let us know if you see anything we might be able to adjust generically.

To learn how Facebook defines and measures key issues, refer to the Community Standards Enforcement Report. The numbers cited in this report are not comparable to the data presented in this RFP as they reference different underlying data. For additional information, see: Community Standards, Enforcement Report Guide.

Finally, we have taken measures to remove URLs and associated engagement statistics that link to known child exploitative imagery from our dataset. We have also taken measures to remove URLs, the 'Title' and 'Blurb' for known non-consensual intimate imagery, suicide and self-harm, although the associated engagement statistics with these links remain in the data set.

References

Mark Bun and Thomas Steinke. *Concentrated differential privacy: Simplifications, extensions, and lower bounds*, pages 635–658. 2016.

John Kelsey, Bruce Schneier, and Niels Ferguson. *Yarrow-160: Notes on the design and analysis of the yarrow cryptographic pseudorandom number generator*, pages 13–33. 1999.

George Marsaglia and Thomas A Bray. A convenient method for generating normal variables. *SIAM review*, 6(3):260–264, 1964.

Linus Torvalds. *Linux Kernel drivers/char/random.c comment documentation.* 2014.