Facebook Privacy-Protected Full URLs Data Set

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

13 February 2020

This dataset is the result of a collaboration between Facebook and Social Science One. It describes the dataset's scope, structure, fields, and privacy-preserving characteristics. This is the second of two planned steps in the release of this "Full URLs dataset", which we described at socialscience.one/blog/update-social-science-one. (We may also have other releases in the future.)

Citation

Messing, Solomon; DeGregorio, Christina; Hillenbrand, Bennett; King, Gary; Mahanti, Saurav; Mukerjee, Zagreb; Nayak, Chaya; Persily, Nate; State, Bogdan; Wilkins, Arjun, 2020, "Facebook Privacy-Protected Full URLs Data Set", https://doi.org/10.7910/DVN/TDOAPG, Harvard Dataverse, V2

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 allowed.

Summary

This document details a data set designed to allow researchers to study the distribution of URLs on Facebook and how users interacted with them. We've protected this data set using differential privacy, which adds enough noise to the data to provide precise guarantees that no significant additional information can be learned from the data about individuals (beyond what is already available from any external source). That means that while no one can learn anything significant about the individuals from the data set (including whether they are in the dataset or not), researchers can still use the data to uncover broad time series or group-level trends and relationships of interest.

The dataset summarizes the demographics of people who viewed, shared, and otherwise interacted with web pages (URLs) shared on Facebook starting January 1, 2017 up to and including July 31, 2019. URLs are included if shared (as an original post or reshare) with "public" privacy settings more than 100 times (plus Laplace(5) noise to minimize information leakage). The URLs have been canonicalized (standardized) and processed (as detailed below) to remove potentially private and/or sensitive data. Aggregate data on user actions on URLs is provided for URLs shared publicly and those shared under the "share to friends" privacy setting.

These data were collected by logging actions taken on Facebook. Logs were processed using a combination of Hive, Presto, and Spark using the Dataswarm job-execution framework. To construct this dataset, we processed approximately an exabyte of raw data from the Facebook platform – including more than 50 terabytes per day of interaction metrics and more than 1 petabyte per day of exposure data (views). The data set has about 38 million URLs, more than half a trillion rows, and more than 10 trillion cell values.

Data from users who have chosen to delete their accounts are not represented in our dataset due to legal constraints, which may have a larger impact on URLs that were shared further in the past (this release is aggregated to provide month-year breakdowns). Users who "deactivate" but do not delete their accounts remain in our dataset.

We have taken measures to remove URLs and associated engagement statistics that link to known child exploitative imagery from these data. 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.

This dataset includes posts from users that have been taken down due to Community Standards violations. 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.

Infrastructure and Resources. Facebook is providing researchers with accepted proposals access to a system that provides an interface to query these data. The interface is a simple SQL layer that will provide access to the tables documented below.

Warnings

1. Privacy-protective procedures have been applied to this dataset (by adding noise in specific ways described below and recoding counts of actions to one when the same user took multiple actions on the same URL). From a statistical perspective, these adjustments induce measurement error, which biases statistical results.[1] Estimates that ignore the error may also induce incorrect coverage for standard errors and thus confidence intervals (in either direction). Researchers at Social Science One are presently conducting work on unbiased statistical approaches for analyzing these data and will release a paper at `SocialScience.One` when available.

2. We do not adjust the privacy-protected data for consistency with other information, as for example the U.S. Census is currently planning in its differentially private releases. As is true any time data contain noise, some values in DP-protected counts will be too large and some too small, and some will be negative, outside the range of the data. However, we are able to make all privacy protective procedures public, which means that researchers can correct for statistical biases. The data may also have other *apparent* inconsistencies, such as fewer views than clicks for some URLs. Appropriate statistical procedures should be used to analyze these data; simplistic approaches like censoring negative values to zero will severely bias statistical estimates.

3. With any large-scale data project, we expect to learn about issues related to data quality, validity, fidelity, etc. in the course of conducting analyses. This is especially the case here, as this is one of the largest social science research datasets ever constructed. Expect issues to arise and let us know what you learn by contacting researchtool-help@fb.com. We hope to respond with fixes to errors fast, where feasible.

[1] The most common bivariate situation is wherein noise biases effects toward zero. However, depending on the quantity of interest, this error may bias estimates in either direction, and in some cases results in estimates with the incorrect sign.

4. These data are considerably larger than commonly used social science datasets, and so will not fit into system memory. Researchers should plan analyses in ways that efficiently use available memory, without exhausting the resources of our computing cluster. At present, this includes limiting system RAM to that of a modern server (e.g., with around 64GB RAM). SQL, or something like it, is required to access these data; Python, Linux, and R will be helpful.

5. The dataset includes 39 major countries. If you need data for your research from a country not included here, please let us know. Your requests will be merged with the country needs highlighted in the proposals from the second round of social science one grantees and prioritized accordingly. Once we get through the second round of grantee countries, Facebook is committed to adding countries to the dataset within one month. Be aware, however, that small countries, and those with a small number of Facebook users, are unlikely to generate informative datasets given the privacy-protective procedures described below.

Data

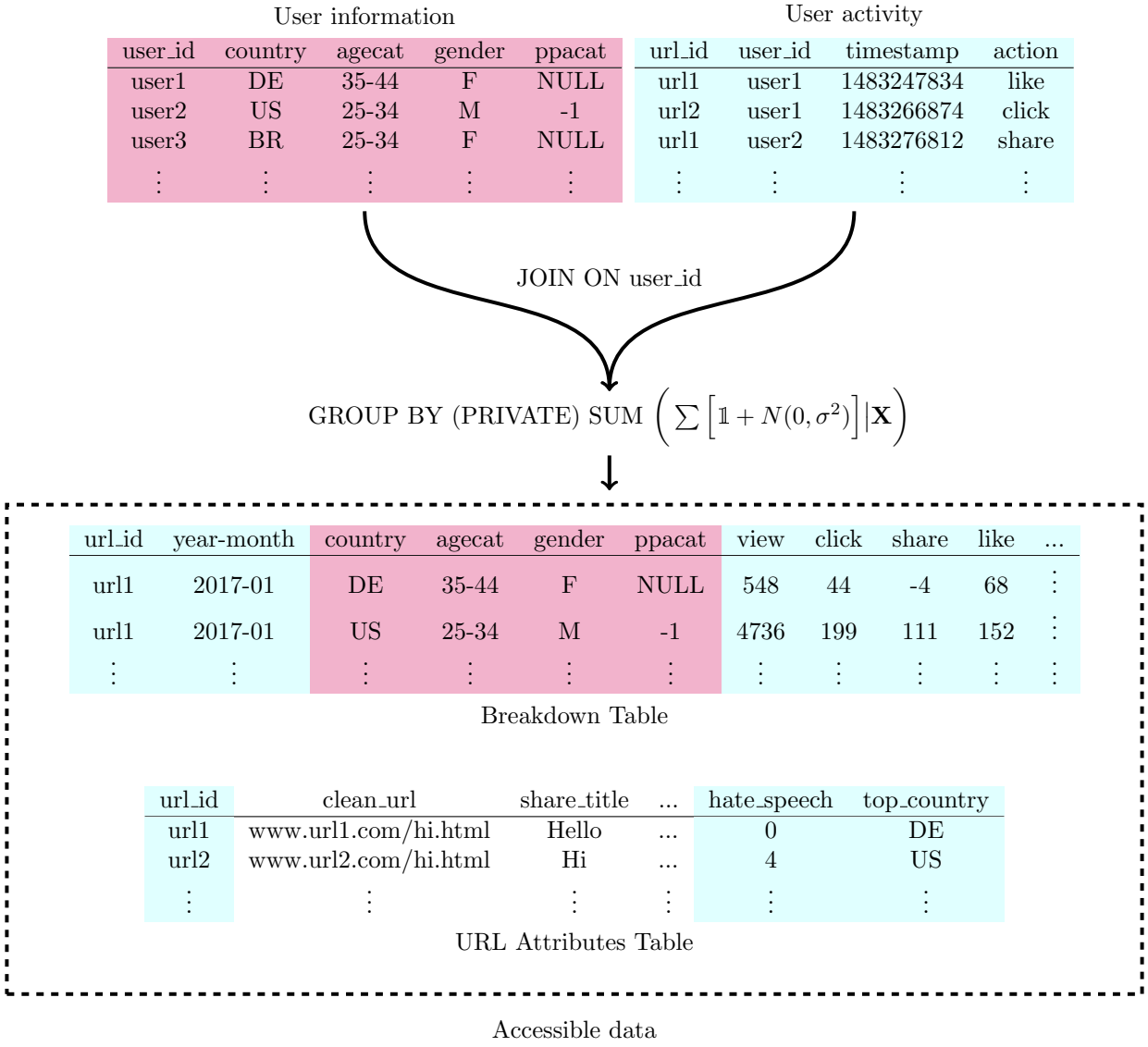
Data Collection and Structure. We include two tables in this release: the *URL Attributes Table* and *Breakdown Table*.

The URL Attributes Table contains the URL-level information that was included in the URLs light release: the URL, domain, timestamp, webpage title and blurb, spam, false news, and hate speech flags, and the country in which it was shared most frequently (after implementing privacy protective procedures). This table includes all URLs "shared publicly" by 100 users (+Laplace(5) random noise) somewhere in the world. "Shared publicly" means that each user chose the public option on their Facebook privacy settings. The table contains 37,983,353 URLs

The Breakdown Table describes counts of users who take particular actions (e.g., views, likes, comments) within URL-year-month-country-age-gender "buckets," for all but the US. This includes 31 year-months, 38 countries, 7 age groups, and 3 gender groups, representing a number of buckets equal to 25,536 times the number of URLs — which is a total of approximately half a trillion, more precisely, 560,913,217,662 rows. Note that this number excludes structural zeros (URLs that haven't yet been shared in a given year-month). For the US, we have 31 year-months, 7 age groups, 3 gender groups, and also have a political page-affinity variable of 5 categories, and so the total number of buckets is 3,360 times the number of URLs — which is approximately 74 billion, or more precisely 73,804,370,745 rows.

The (total 634,717,588,407) rows of the breakdown table are these buckets. Each unique user-URL-action is represented by the number 1, which is added to precisely one of the buckets. This means that if a user shares a URL more than once, only the first share is counted; this variable can be thought of as either the number of users who take a specific action or the total number of actions re-coded to one per user. This means that metrics related to unique user-URL-actions in these data will not match other data sources that provide counts of total shares, as some Facebook users can and do take the same action on a URL repeatedly.) We define a "user" throughout this codebook as a unique Facebook account (even though some people may have more than one Facebook account and some may share accounts).

This table was created by joining two tables of disaggregated data at Facebook: a user information table and a user-activity table (see figure below). The resulting user activity and demographic breakdowns table contains counts—to which Gaussian noise is added so as to provide differentially privacy—of various actions broken down by country, age brackets, gender, and, in the U.S., political page affinity brackets (defined in additional detail below).



The "URL Attributes" and "Breakdown" tables can be joined using a "url_id" key. Aggregate counts have been protected by adding noise, as per the 'privacy protection' section below. Rows that contain 0 counts prior to adding noise have not be removed to ensure that we are not leaking information about rows containing 0 entries.

Variables Aggregate statistics in the breakdown table marked "DP" have noise added for differential privacy. An artificial example dataset with a few observations can be found at https://bit.ly/FullURLsEG; it may be helpful in understanding the fields described below.

URL Attributes Table

- **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, not just the domain (e.g., https://www.nytimes.com/2018/07/09/world/asia/thailand-cave-rescue-live-updates.html).

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. Our post-processing procedure is explained below.

- **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 are 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).
- **3pfc_rating [text]:** If URL was sent to third-party fact-checkers (3pfc), 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.
- **3pfc_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.
- **3pfc_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.
- **spam_usr_feedback**** **[integer]**: the total number of unique users who reported posts containing the URL as spam.
- **false_news_usr_feedback**** **[integer]**: the total number of unique users who reported posts containing the URL as false news.
- **hate_speech_usr_feedback**** **[integer]**: the total number of unique users who reported posts containing the URL as hate speech.
- **public shares top country [text]**: URL shares are tallied by country and the country with the most (differentially private) 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.

Breakdown Table

Three types of variables can be found here: the *keys* that define the row-units (as the Cartesian product of all keys); a *unique URL id*, which is a URL-level observation that uniquely identifies a URLs (and can be linked to from the URLs Attributes Table); and the *aggregate statistics*, which are summaries (e.g., sum or mean) within, i.e., conditioned on or grouped by each of the keys. The data contain aggregated counts of the number of users who have shared, viewed, clicked, liked (or otherwise reacted), or commented on each URL.

- **Url_rid [text]**: a unique URL id created specifically for this data set.
- **Keys**: These variables define the "buckets" which are the rows of this table.
  - **Year-month [string]:** year and month. Data will be partitioned on this variable.
  - **Country [text]**: the country in which the actions below occur. This variable is stored in a column called **c** in the actual dataset. Data will be partitioned on this variable and due to considerations related to the size of the data, for now we are prioritizing the release of all countries in researchers' proposals, though we are in the process of adding more. This release will contain data for countries needed to conduct analysis for research proposals already approved through Social Science One — Brazil, Canada, Chile, Hong Kong, Israel, Kenya, Switzerland, Taiwan, the United States, Zimbabwe, and the 28 EU Member states[2] (Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Italy, Ireland, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, and the United Kingdom).
  - **Age_bracket [text]:** Age brackets include: 18-24, 25-34, 35-44, 45-54, 55-64, 65+, NA. Data from users' profiles.
  - **Gender [text]:** male, female, other. Data from users' profiles.
  - **Political page-affinity [integer, ordered]:** These buckets [-2, -1, 0, 1, 2] take the pages people follow and scale them into components associated with political affinity, based on Barberá et al. [2015]. This variable is only available in US data.
- **Aggregate statistics:** these columns contain aggregated URL-user-actions, counts of the number of people who fall into the bucket corresponding to the cell of the cross-products of the key variables in question.
  - **views [integer], DP:** Number of users who viewed a post containing the URL.
  - **clicks [integer], DP:** Number of users who clicked on the URL.
  - **shares [integer], DP:** Number of users who shared the URL in a post or reshared such a post.
  - **total_share_without_clicks, DP [double]**: Number of users who shared a post containing the URL but did not actually click on the link. (Some users share articles without first clicking through to the actual content. Hence, this number may help identify articles that users are sharing without reading, or URLs used in organized campaigns to spread content.)
  - **comments [integer], DP:** Number of users who comment on posts containing the URL
  - **likes [integer], DP:** Number of users who liked posts containing the URL
  - **loves [integer], DP:** Number of users who 'love' posts containing the URL
  - **hahas [integer], DP:** Number of users who react 'haha' posts containing the URL
  - **wows [integer], DP:** Number of users who 'wow' posts containing the URL

[2]There will be a European Union-level aggregation in the data, constructed by summing the country-counts across buckets after noise.

- **sorrys [integer], DP:** Number of users who 'sad' posts containing the URL. Note that the official name for this reaction is 'sad', but the column name in this dataset is 'sorrys.'[3]
- **angers [integer], DP:** Number of users who 'angry' posts containing the URL

**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.

Privacy Protection

We are releasing aggregates in these data using a privacy preserving technology called *differential privacy*, which allows researchers to uncover trends and patterns in the data without learning about the behavior of specific individuals. More precisely, differential privacy enables us to provide precise guarantees that no significant *additional* information about an "action" on Facebook (such as sharing a URL or liking it) taken by a person can be learned from the data beyond what is already available from external sources. This guarantee is quantified by a precise mathematical bound. Unlike other privacy protecting technologies, such as (attempts at) de-identification, differential privacy guarantees hold regardless of the auxiliary data and computing resources an adversary may possess.

Differential privacy provides *plausible deniability* to people whose information is included in the data set. In this case, it's impossible to determine—in a way that is significantly better than random guessing—whether a specific user took an action in these data, because differential privacy makes it impossible to isolate a specific row. That means it is impossible to determine whether or not information about the action exists in the dataset at all—again, in a way substantially better than random guessing, and where "substantially" is precisely quantified by a privacy parameter.

The privacy guarantees we are providing with this data release go another step and protect not only each action by user, but also all the actions by an individual user considered together (for 99% of users). This means it's not possible to determine whether or not all but the most active 1% of individuals are represented in the data at all. This guarantee may seem less important because such a high proportion of people have Facebook accounts, but even if all people have Facebook accounts being represented in our data requires having interacted with a URL shared at least approximately 100 times, and that fact is not public.

These privacy guarantees of both actions and users are generally operationalized by adding noise to data or the results of statistical procedures, or censoring large values to a fixed range. A non-technical introduction to differential privacy is available here: http://privacytools.seas.harvard.edu/files/privacytools/

[3]For more details on Facebook Reactions, see the following post from the Facebook Newsroom.

files/pedagogical-document-dp_0.pdf; a rigorous introduction can be found here: https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf. Approaches to differential privacy that provides statistical guarantees for researchers can be found at Evans et al. [2020] and Evans and King [2020], where the latter offers methods designed to avoid the biases in this dataset in particular.

We now explain how we guarantee action-level differential privacy followed by how we guarantee user-level differential privacy.

Action-level zero-Concentrated Differential Privacy (zCDP). In this release, data aggregates that describe actions taken by a user on a URL are protected under a form of differential privacy called zero-concentrated differential privacy (zCDP, see Bun and Steinke [2016]).

The privacy protections are "action level" (rather than the more familiar "user-level" differential privacy) in that the granularity of what is protected is not a single user, but rather a single action, or user-URL-action (e.g. a user sharing a specific URL).

We have made choices about the data that allow us to add significantly less noise compared with other formulations of differential privacy, essentially because it is easier to hide a great many modest numbers than a few large numbers. First, we define the unit of analysis as the *unique* user-URL-action tuple, which can occur in the data only once. This generally amounts to de-duplicating actions taken in the data. For example, if a user liked a post with the same URL more than once, we take the first instance and discard others.

This allows us to take advantage of assumptions underlying zCDP that achieve differential privacy guarantees that entail adding significantly less noise with minimal impact on the data. Because zCDP relies on the l_2 norm to formulate sensitivity (in this case the square root of the sum of squared indicator variables), we can add significantly lower levels of noise than if we instead allowed an arbitrary number of actions per user and relied on other variants of differential privacy with l_1 sensitivity formulations.

Formal definition of zCDP from Bun and Steinke [2016] Under zCDP, the key parameter governing privacy and thus noise is ρ. This parameter can be used to bound a user-level ε, the privacy parameter in the more standard parameterization of (ε, δ)-differential privacy. Below we provide a formal definition of zCDP including how the privacy parameter ρ relates to the Gaussian noise to be added to the data set, governed by σ. We then work backwards, setting ρ to attain an approximate user-level ε of 0.45 for each column in the data set.

A random mechanism, $M : \mathcal{X}^n \to \mathcal{Y}$ is (ρ)-zero-concentrated differentially private[4] if, for all $x, x' \in \mathcal{X}^n$ differing on a single entry and all $\alpha \in (1, \infty)$:

$$\mathrm{D}_\alpha \left(M(x) \| M\left(x'\right) \right) \leq \rho\alpha$$

where $\mathrm{D}_\alpha \left(M(x) \| M\left(x'\right) \right)$ is the α-Renyi divergence (see Van Erven and Harremos [2014] for a definition and comparison to KL divergence) between the distribution of $M(x)$ and the distribution of $M\left(x'\right)$.

Define a privacy loss function such that the privacy loss between two random variables Y and Y' is given by a new variable, $Z = Privloss\left(M(x) \| M\left(x'\right) \right)$. Z is then distributed according to $f(Y)$, where $f(y) = \log\left(\mathbb{P}[Y = y] / \mathbb{P}\left[Y' = y\right] \right)$, where all randomness in this distribution is due to the randomness in the mechanism, not a hypothetical data generating process.

The inequality above can then be re-written as a bound on the moment generating function of the privacy loss:

[4] We are using a special case of (ξ, ρ)-zero-concentrated differentially privacy by setting $\xi = 0$.

$$\mathbb{E}\left[e^{(\alpha-1)Z}\right] \leq e^{(\alpha-1)(\rho\alpha)}$$

The fact that zCDP entails a bound on the moment generating function of the privacy loss Z, $\mathbb{E}\left[e^{(\alpha-1)Z}\right]$ means that Z resembles a Gaussian distribution with mean ρ and variance 2ρ. This implies:

$$\mathbb{P}[Z > \lambda + \rho] \leq e^{-\lambda^2/4\rho}$$

for all $\lambda > 0$.

To define sensitivity: a function $q : \mathcal{X}^n \to \mathbb{R}$ is said to have sensitivity Δ if for all $x, x' \in \mathcal{X}^n$ differing on only a single entry, $|q(x) - q(x')| \leq \Delta$.

If $M(x)$ produces a sample from $N\left(q(x), \sigma^2\right)$, then M satisfies $\left(\Delta^2/2\sigma^2\right)$-zCDP.

The inequalities defining zCDP are *exactly tight* for the Gaussian mechanism for all values of α. For more details, see Bun and Steinke [2016].

Under zCDP, the parameter summarizing the privacy guarantee is ρ, which is achieved using the Gaussian mechanism. For the count queries here, the action-level sensitivity $\Delta = 1$ (user-level sensitivity varies as explained below). In other words, the mechanism entails adding $N(0, \sigma^2)$ noise to the data, where the relationship between σ and ρ follows (see also Bun and Steinke [2016]):

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

We can translate this to the more familiar epsilon-delta differential privacy framework for ease of interpretation. We use the following (Lemma 3.6, Bun and Steinke 2016):

$$\varepsilon = \rho + \sqrt{4\rho \log(\sqrt{\pi\rho}/\delta)}$$

or

$$\varepsilon = \frac{1}{2\sigma^2} + \sqrt{\frac{2}{\sigma^2} \log\left(\frac{\sqrt{\pi}}{\delta\sqrt{2\sigma^2}}\right)}$$

Because zCDP as defined here relies only on ρ, the translation to (ε, δ)-differential privacy is a two-dimensional surface, so ε depends on δ and vice-versa. If we know $\rho = 0.005$, we still must select a value for δ to get ε. So if $\rho = 0.005$, we can set $\delta = 1 \times 10^{-5}$, yielding $\varepsilon = 0.485$. Or, we can set $\delta = 1 \times 10^{-6}$, yielding $\varepsilon = 0.531$.

We can then think of users as "groups of actions" and examine how action-level privacy relates to user-level privacy, by relating the privacy guarantee on a single action to the privacy guarantee on a group of actions.

Relation to user-level differential privacy. We formulate an analogy to user level privacy based on this action-level definition. As users take more unique URL-actions (e.g., they click on different URLs), their total contribution to the data grows and more noise is required to provide plausible deniability that the user ever appeared in the data set—or in other words to provide user-level differential privacy. To be

more precise, users are protected under a differential privacy guarantee similar to user-level privacy if they've taken at most k url-actions, which relies on the fact that a user-URL-action can occur only once.

We map action-level privacy to user-level privacy in two steps. First, for any one of the 11 possible actions (listed under aggregate statistics) a user may taken an arbitrary number of actions—for example, a user may have clicked on many hundreds of URLs in our dataset, a quantity we define as k. Second, we also wish to protect the fact that a user may take up to 11 of these possible actions on any one URL.

Using the composition properties of zCDP[5], we can compute the privacy guarantee for a group of k actions by a user. The l_2 sensitivity for any user who has taken at most k *unique* url-actions for count queries is $\sqrt{\sum_1^k 1^2} = \sqrt{k}$ (see Dwork and Roth [2014] for a review of l_2 sensitivity and its relationship to the Gaussian mechanism versus l_1 sensitivity and the Laplacian mechanism). We can then offer a group-level ρ-zCDP guarantee for a group of size k by adding a sufficient amount of noise to satisfy:

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

thus satisfying group-level zCDP (see Proposition 1.9, Bun and Steinke [2016], page 7-9). We can combine this equation with Lemma 3.6, Bun and Steinke [2016] to provide a user-level privacy equivalent ε:

$$\varepsilon = \frac{k}{2\sigma^2} + \sqrt{\frac{2k}{\sigma^2}\log\left(\frac{1}{\delta}\cdot\sqrt{\frac{k\pi}{2\sigma^2}}\right)}$$

To solve for σ, we can begin by reformulating the equation above as:

$$\frac{\sigma^2\epsilon^2}{2k} + \frac{k}{8\sigma^2} + \log\sigma - \frac{\epsilon}{2} - \log\left(\frac{1}{\delta}\right) - \frac{1}{2}\log(k\pi) - \frac{1}{2}\log(2) = 0$$

We can solve for σ using Newton's method, which involves picking a candidate value for σ and iterating over the equation $\sigma_{i+1} = \sigma_i - \frac{f(\sigma_i)}{f'(\sigma_i)}$ until σ converges at the root. We define $f(\sigma)$ and $f'(\sigma)$ as follows:

$$\begin{aligned} f(\sigma) &= \frac{\sigma^2\epsilon^2}{2k} + \frac{k}{8\sigma^2} + \log(\sigma) - \frac{\epsilon}{2} - \log\left(\frac{1}{\delta}\right) - \frac{1}{2}\log(k\pi) - \frac{1}{2}\log(2) \\ f'(\sigma) &= \frac{\sigma\epsilon^2}{k} - \frac{k}{4\sigma^3} + \frac{1}{\sigma} \end{aligned}$$

For example, if we wish to protect users who have taken 100 unique [url-]actions ($k = 100$) with a formal (user-level) ε guarantee of 0.45, we can set delta to $\delta = 10^{-5}$, and add $N(0, \sigma = 98)$ noise.[6]

If a user actually made $k' > k$ [url-]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 above wherein $k = 100$, if for a given person, $k' = 150$, this person's effective ε will not be approximately 0.5, but instead will be 0.75. If $k' = 200$, effective ε will be 1, etc (still assuming weve fixed δ at 10^{-5}). We've set k such that 99% of users are protected under the user-level differential privacy guarantee.

[5] We could obtain a weaker bound using the generic "group privacy" guarantee of zCDP, but in this instance, stronger bounds are possible using composition properties. This is because users can affect each count by at most 1, which results in a group of k actions having total l_2 sensitivity of $\sqrt{k}$, rather than k.

[6] Note that selecting different values of delta will change the translation—for example, we can set delta to $\delta = 10^{-6}$ with the same user-level $\varepsilon = 0.5$ and get $N(0, \sigma = 107)$; or add the same amount of noise, $N(0, \sigma = 98)$, and achieve a user-level $\varepsilon = 0.542$

In the table below, we define how much noise is added to each variable. First, for each user, we count number of unique actions each user takes on each URL. For each action, we then set k at a value above the (differentially private) 99th percentile of unique url-actions taken.[7] [8] We then select ρ to ensure our final user-level $\varepsilon_{\text{user}}$ parameter is under .45. The noise parameter σ follows from the equations above, as does our [url-]action-level ε parameter, after setting $\delta = 10^{-5}$.

For example, for shares, we first compute the total number of URLs shared by each user. We then compute the differentially private 99th percentile and round up to the nearest positive integer to get k. We then plug k, $\varepsilon_{\text{user}} = 0.45$, and $\delta = 10^{-5}$ into the equations above to solve for ρ and thus σ.

Action	k	ρ	ρ_{user}	ε	$\varepsilon_{\text{user}}$	δ	σ	Pct users protected
views	51914	0.0000	0.0052	0.00	0.45	10^{-5}	2228	> 99
click	17	0.0003	0.0052	0.10	0.45	10^{-5}	40	> 99
share	2	0.0026	0.0052	0.31	0.45	10^{-5}	14	> 99
share_without_click	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
comment	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
like	5	0.0010	0.0052	0.19	0.45	10^{-5}	22	> 99
angry	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
haha	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
love	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
sad	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
wow	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
false_news_usr_feedback	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
hate_speech_usr_feedback	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
spam_usr_feedback	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99

To provide a sense of how many users are protected under differential privacy at the $\varepsilon_{\text{user}} = 0.5$ level in the full table, we compute user-level multivariate histograms for all actions and compute the proportion of users whose url-action counts are all uniformly lower than all k for each action in the vector $\boldsymbol{k}$, from the table above. That number is 96.6 percent. Note that the total privacy consumption (setting $\delta = 10^{-5}$) is *not* simply $\varepsilon_{\text{user}} = 0.5 \cdot 14 = 7$, but rather

$$\varepsilon_{\text{user}} = \sum \boldsymbol{\rho} + \sqrt{4 \sum \boldsymbol{\rho} \cdot \log(\sqrt{\boldsymbol{\rho} * \pi}/\delta)} = 1.844$$

Implementation. We operationalize this protection simply by adding Gaussian noise to the aggregations (counts) based on the table above.

We generate this noise using the Yarrow-160 cryptographically secure pseudo-random number generator [Kelsey et al., 1999]. Our implementation relies on noise generated from the /dev/urandom device in the Linux kernel. The idea is to gather "environmental noise," including inter-keyboard timings, inter-interrupt timings from some interrupts, and other non-deterministic events that are difficult for an adversary to measure [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.

[7] For this calculation we use noisy-min, (see Dwork and Roth [2014]). For each percentile p, we define $f(v)$ as the proportion of people who have less than v actions and compute $s_v = |f(v) - p| + \text{Lap}(2/(\varepsilon_{\text{user}} \times N))$ and take $\underset{v}{\text{argmin}} s_v$. We set $\varepsilon_{\text{user}}$ to 0.001. N here is Facebook's monthly active user base at the end of data collection, approximately 2.4 billion.

[8] For the view field, we estimated the 99th percentile by taking the percentile across 4 randomly selected weeks (2017-05-15 - 2017-05-23; 2018-04-09 - 2018-04-17; 2018-09-25 - 2018-10-02; 2019-04-30 - 2019-05-07) and multiplying by the ratio of days in our data to days in our sample, 33.82.

Other privacy-preserving technology. We have applied other privacy-preserving technologies to these data in addition to differential privacy. First, access is limited to grantees and provided in a secure environment. What's more, all URLs included in these data have been shared at least 100 times $+Lap(5)$ noise by unique users with fully public privacy settings and we've taken steps to remove unintentional PII from URLs and ensure they contain only navigation-critical information as outlined below.

URL Sanitization

This section describes the URL-sanitization procedures used to clean the data set. The code used to execute steps 3-8 below can be found here: https://github.com/facebookresearch/URL-Sanitization.

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. NOTE: the terminal webpage may differ from the "og:url" tag. For example: the "og:url" tag for https://www.dailymail.co.uk/news/article-4367746/WikiLeaks-says-CIA-disguised-hacking-Russian-activity.html is actually http://www.dailymail.co.uk/~/article-4367746/index.html, which is how the URL is recorded in this data set.[9] Researchers can use Facebook's Object Debugger https://developers.facebook.com/tools/debug/og/object/ for information about any single URL. Furthermore, due to a number of prominent websites including the Washington Examiner and FoxNews.com making changes to their websites, the og:url tag will sometimes point to a different URL today than when it was originally shared on Facebook.[10] We provide the originally shared canonical URL.
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 (∼21.8% 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.

[9] Thanks to Simon Hegelich for surfacing this example.

[10] We thank Juan Carlos Medina Serrano for pointing this out. One example includes the URL http://www.washingtonexaminer.com/a-hillary-clinton-donor-paid-500000-to-fund-women-who-would-accuse-trump-of-sexual-misconduct/article/2644747 which currently resolves to the following address https://www.washingtonexaminer.com/a-hillary-clinton-donor-paid-500-000-to-fund-women-who-would-accuse-trump-of-sexual-misconduct. However, previous versions of the web page resolved to the former address and used that address in the "og:url" meta-tag, as can be seen via the Wayback Machine: https://web.archive.org/web/20180104233644/http:/www.washingtonexaminer.com/a-hillary-clinton-donor-paid-500000-to-fund-women-who-would-accuse-trump-of-sexual-misconduct/article/2644747.

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. Any email addresses that appear in any part of the URL string are removed using regular expressions.

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

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.

Acknowledgements: We thank Aaron Roth and Ilya Mironov for providing extensive guidance on differential privacy. We also thank Danfeng Zhang, Salil Vadhan, Abhradeep Guha Thakurta, Thomas Steinke, Julia Lane, Daniel Goroff, Daniel Kifer, and Cynthia Dwork for feedback and helpful discussions.

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

Appendix 1: Modified Report Noisy Max

Below, we reproduce an analysis based on a extension of the "report noisy max" differential privacy mechanism from Dwork and Roth (2014) to arbitrary sensitivity Δ score functions. This algorithm is based on correspondence with Aaron Roth and reproduced here with his permission.

Let $\mathcal{X}^n$ be an arbitrary data domain and let $\mathcal{O}$ be an arbitrary finite outcome space of cardinality K. Let $f : \mathcal{X}^n \times \mathcal{O} \to \mathbb{R}$ be an arbitrary sensitivity Δ function in its first argument (i.e. $f(\cdot, o)$ is a sensitivity Δ function for all $o \in \mathcal{O}$). Define *report noisy max* as the algorithm that first samples $Z_o \sim \text{Lap}(2\Delta/\varepsilon)$, and then outputs $RNM(D) = \arg\max_{o \in \mathcal{O}}(f(D, o) + Z_o)$.

Theorem 1. *The Report Noisy Max algorithm satisfies ε-differential privacy.*

Proof. To simplify notation, throughout the argument, we assume that $\arg\max_{o \in \mathcal{O}}(f(D, o) + Z_o)$ is unique. This is true with probability 1 over the randomness of Z, and hence does not affect the claim of differential privacy. Given a noise vector $Z \in \mathbb{R}^K$, write $o(D, Z) = \arg\max_{o \in \mathcal{O}}(f(D, o) + Z_o)$ to denote the element output by report noisy max, given that the noise is realized as Z. For each $D \in \mathcal{X}^n$ and $o \in \mathcal{O}$, Let $\mathcal{E}(D, o) = \{Z : o(D, Z) = o\}$ be the set of noise vectors that result in o being output. Fixing any output $o \in \mathcal{O}$ and noise vector $Z \in \mathbb{R}^K$, let $\tilde{Z}$ be the vector such that $\tilde{Z}_o = Z_o + 2\Delta$, and $\tilde{Z}_{o'} = Z_{o'}$ for all $o' \neq o$.

First, observe that by inspection of the pdf of the Laplace distribution, when each coordinate is sampled independently $Z_o \sim \text{Lap}(2\Delta/\varepsilon)$ we have that (abusing notation to write $\Pr[Z]$ for the probability density of the vector Z) $\Pr[Z] \leq e^{\varepsilon} \Pr[\tilde{Z}]$.

The crux of the argument follows from the following lemma:

Lemma 1. *For every pair of neighboring $D, D' \in \mathcal{X}^n$, $Z \in \mathbb{R}^K$, and $o \in \mathcal{O}$, if $Z \in \mathcal{E}(D, o)$ then $\tilde{Z} \in \mathcal{E}(D', o)$. In particular:*

$$\mathbb{1}[Z \in \mathcal{E}(D, o)] \leq \mathbb{1}[\tilde{Z} \in \mathcal{E}(D', o)]$$

Proof. Observe that for every $o' \neq o$, we have:

$$\begin{aligned} f(D', o) + \tilde{Z}_o &\geq f(D, o) - \Delta + \tilde{Z}_o \\ &= f(D, o) + Z_o + \Delta \\ &> f(D, o') + Z_{o'} + \Delta \\ &\geq f(D', o') + Z_{o'} \\ &= f(D', o') + \tilde{Z}_{o'} \end{aligned}$$

Here the first and last inequalities follow from the fact that f is a sensitivity Δ function in its second argument, and the third inequality follows from the fact that o is the unique maximizer of $f(D, o) + Z_o$. Hence we can conclude that o is also the unique maximizer of $f(D', o) + \tilde{Z}_o$. □

Again abusing notation by writing $\Pr[Z]$ to refer to the probability density on the vector Z, we can calculate:

$$\begin{aligned}
\Pr[RNM(D) = o] &= \int_{\mathbb{R}^K} \Pr[Z] \cdot \mathbb{1}[Z \in \mathcal{E}(D, o)] dZ \\
&\leq \int_{\mathbb{R}^K} e^{\varepsilon} \Pr[\tilde{Z}] \cdot \mathbb{1}[Z \in \mathcal{E}(D, o)] dZ \\
&\leq \int_{\mathbb{R}^K} e^{\varepsilon} \Pr[\tilde{Z}] \cdot \mathbb{1}[\tilde{Z} \in \mathcal{E}(D', o)] dZ \\
&= \int_{\mathbb{R}^K} e^{\varepsilon} \Pr[Z] \cdot \mathbb{1}[Z \in \mathcal{E}(D', o)] dZ \\
&= e^{\varepsilon} \Pr[RNM(D') = o]
\end{aligned}$$

Here the third line follows from Lemma 1 and the fourth line follows from a change of variables. This establishes ε-differential privacy. □

Appendix 2: Distribution of URL-interactions across countries and variables.

In order to provide more insights into the distribution of the number of unique URL actions taken by country, we calculated differentially private percentiles, using the noisy-min method discussed in footnote 6 and Appendix 1. These are available on the Facebook Research Tool.

References

Pablo Barberá, John T Jost, Jonathan Nagler, Joshua A Tucker, and Richard Bonneau. Tweeting from left to right: Is online political communication more than an echo chamber? *Psychological science*, 26(10): 1531–1542, 2015. URL https://journals.sagepub.com/doi/abs/10.1177/0956797615594620.

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

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Georgina Evans and Gary King. Statistically valid inferences from differentially private data releases. 2020. URL GaryKing.org/dpd.

Georgina Evans, Gary King, Margaret Schwenzfeier, and Abhradeep Thakurta. Statistically valid inferences from privacy protected data, 2020. URL GaryKing.org/dp.

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.

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

Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.