Preserving Anonymity in Data Collection

The Story of Differential Privacy

Alasdair Kergon <agk@redhat.com>

Pavel Odvody <podvody@redhat.com>

June 13-15, 2024

DEVCONF.CZ, Brno

Motivation

- Scenarios where you are gathering statistical data
- Scenarios where you must preserve the privacy of individual contributors and in principle be unable to identify them
- Differential Privacy provides a sophisticated mathematical framework for calculating the trade-off between accuracy and detail/privacy
- Gather telemetry data about software usage without identifying individuals
- Reduce risk of leaking any exact data used to train an AI model

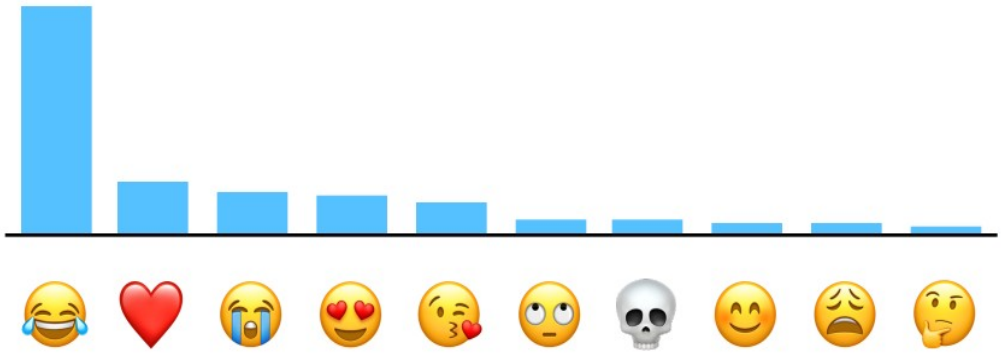
fedora™

Formal Advantages

- Works independently of your attack models
- Places an upper bound on the amount of information revealed to any attacker

Existing Usage

- Apple
  - Emoji suggestions
  - Domains crashing Safari
- Microsoft
  - SmartNoise
- Many more examples at https://desfontain.es/blog/real-world-differential-privacy.html



The Count Mean Sketch technique allows Apple to determine the most popular emoji to help design better ways to find and use our favorite emoji. The top emoji for US English speakers contained some surprising favorites.

Mechanisms

- De-identification
  - Remove all unique identifiers e.g. names, phone numbers
- k-Anonymity and l-Diversity
  - All attributes that might identify an individual are present in the data for at least k individuals
  - Each grouping of individuals must contain at least l different data values
- Differential Privacy
  - When a statistical procedure is run over privacy-sensitive data, the output of the process would not change noticeably if any arbitrary individual's data had been omitted from the dataset being used
  - Restriction applies to the process not the data

DP Scenarios

- Query central database
- Add noise
  - Locally
    - At scale, when there will be many systems with the same data, it may only be necessary to collect a sample of carefully-hashed data bits from each client. (E.g. Bloom filters.)
  - AI Model training data
  - AI Model weights
- Trusted Third Party
  - Two-level

Noise

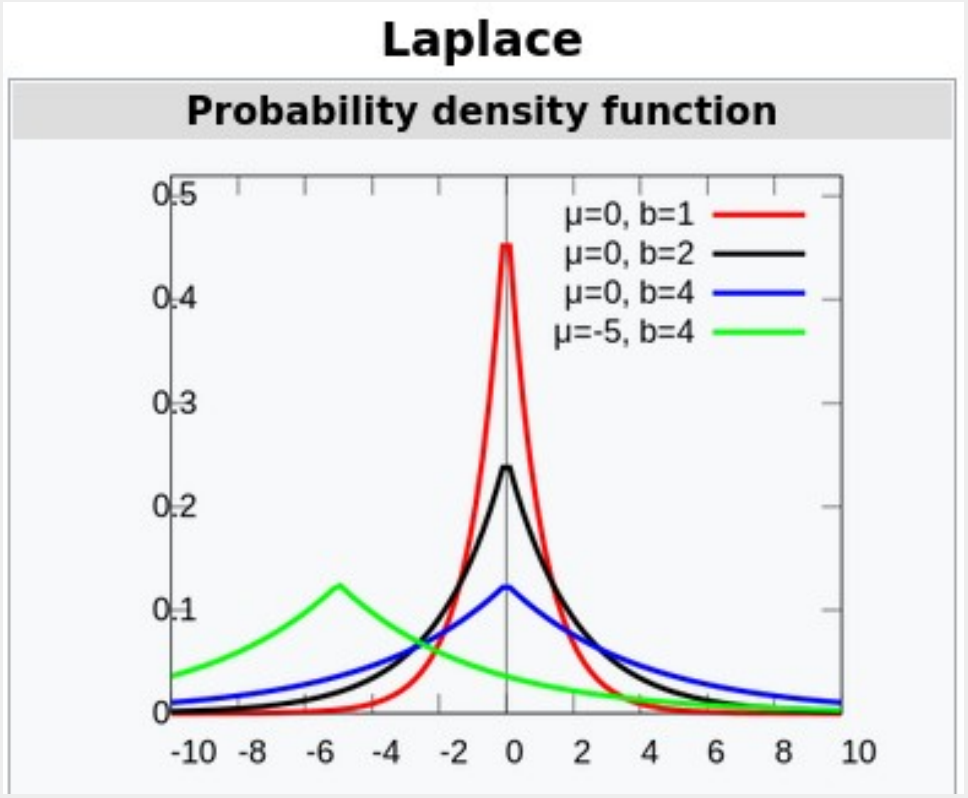
- Random small adjustments
  - Minimal impact on statistics
  - Masks data relating to specific individuals
- Average characteristics of noise known
  - Subtract from aggregate, leaving only data

Noise – Coin Toss

- Every contributing user has a parameter that is either 0 or 1 and we want to know their relative proportions
- Each user tosses a coin
- If heads, send the actual value of 0 or 1; if tails, send a random 0 or 1
- On average, 50% of the data will be random so subtract out 25% of 0s and 25% of 1s to obtain the distribution
- You never know if the data from any individual user was real or random

Noise Types

- Laplace distribution (Double exponential)
- Used to adjust categorical data



(wikipedia)

fedora™

3rd Party Aggregation

- Every contributing user possesses an integer parameter between 1 and 100 inclusive which we want to average
- We employ two independent third parties that everyone trusts not to collude with each other
- The third parties each add up the numbers the users send to them and publish the total
- Users pick a random integer between 1 and 100 and they send one third party their number with the random integer added (modulo 100), and send the other their number with it subtracted
- If we combine the published totals, all the random numbers cancel out, as they were both added and subtracted, so we can get the average value
- No deductions can be made about any individual's value as a random distribution shifted by an integer remains a random distribution

Privacy Budget

- Defines the maximum tolerance in the output for revealing information about the input data
- Keeps the total amount of information revealed within an acceptable range
- Tracked across multiple operations so repeated queries cannot be used to bypass privacy restrictions
- Limit the number of contributions to the output from any individual user, otherwise, because noise averages out, some information about an individual might be deduced

Privacy Loss Parameters

- Traditionally represented by Greek letters Epsilon (ε) and Delta (δ)
- ε determines amount of noise added to the data
- Specifically measures the effect of one individual's information on the output.
- The probability of any specific output with all the data compared against the probability of that same output with any one individual excluded from the data is bounded by a function of ε

  Low value means strong privacy protection but low accuracy. (If ε is 0 the output is meaningless.)
- δ accounts for the probability that the privacy guarantee might be breached

Sensitivity Parameter

- The Sensitivity (Δf) measures the greatest effect that the input from a single individual can have on the output.
- Determines how much noise needs to be added
- Laplace noise is proportional to Δf and inversely proportional to ε

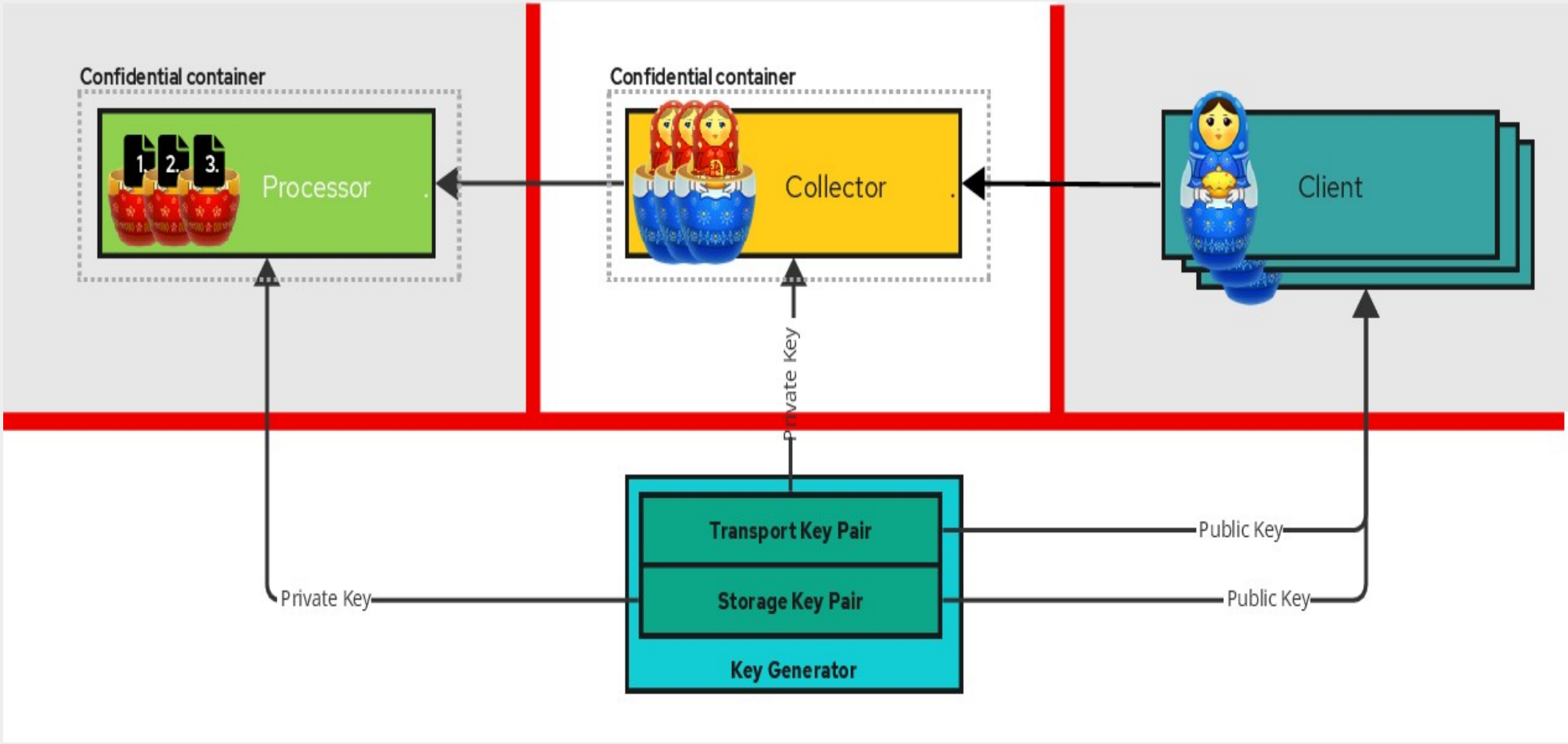
Open Source Libraries

- We evaluated three libraries for potential use with our distributions
  - Harvard/Microsoft (and others) - OpenDP - https://opendp.org/
  - Google - https://github.com/google/differential-privacy
  - IBM - https://diffprivlib.readthedocs.io/en/latest/
- There is no time to cover the details, but we decided that OpenDP was the best all-round fit. We were impressed by its documentation, architecture and strong academic underpinnings. Its biggest shortcoming was not unusual and in an area where we are already experts – packaging.

Fedora

- Desktop developers wish to gather telemetry covering hardware, software, disks, performance, codecs, apps etc.
- https://fedoraproject.org/wiki/Changes/Telemetry
- https://etherpad.opensuse.org/p/metrics-list
- We are putting forward these techniques of Differential Privacy as potential mechanisms to help them to achieve those aims

Confidential Telemetry



fedora™

Demo

- Download Fedora client package
- Collects uptime and list of Fedora packages
- Wraps with two layers of encryption
- Sends to remote container for aggregation
- https://gitlab.com/confit-project/confit
- https://people.redhat.com/agk/talks/devconf_2024/diffpriv_devconf24_demo.mp4

Additional References

- These slides – https://people.redhat.com/agk/talks/devconf_2024/diffpriv_devconf24.pdf
- Original paper “Calibrating Noise to Sensitivity in Private Data Analysis” (2006) – https://doi.org/10.1007/11681878_14
- Comprehensive introduction – https://desfontain.es/blog/friendly-intro-to-differential-privacy.html
- Apple –https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf
- Microsoft –https://blogs.microsoft.com/ai-for-business/differential-privacy/
- MIT – Practical Handbookhttps://admindatahandbook.mit.edu/book/v1.0/diffpriv.html
- DevConf – Gordon Haff https://www.youtube.com/watch?v=VrliEn9ktmE https://www.youtube.com/watch?v=aVnBfi3CJPA

Advanced techniques

- RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response (2014, Google)
- Randomized Aggregatable Privacy-Preserving Ordinal Response, or RAPPOR, is a technology for crowdsourcing statistics from end-user client software, anonymously, with strong privacy guarantees. In short, RAPPORs allow the forest of client data to be studied, without permitting the possibility of looking at individual trees. By applying randomized response in a novel manner, RAPPOR provides the mechanisms for such collection as well as for efficient, high-utility analysis of the collected data. In particular, RAPPOR permits statistics to be collected on the population of client-side strings with strong privacy guarantees for each client, and without linkability of their reports. This paper describes and motivates RAPPOR, details its differential-privacy and utility guarantees, discusses its practical deployment and properties in the face of different attack models, and, finally, gives results of its application to both synthetic and real-world data.
- https://research.google/pubs/rappor-randomized-aggregatable-privacy-preserving-ordinal-response/

Implementation Issues

- Widespread Underestimation of Sensitivity in Differentially Private Libraries and How to Fix It (2022)
- We identify a new class of vulnerabilities in implementations of differential privacy. Specifically, they arise when computing basic statistics such as sums, thanks to discrepancies between the implemented arithmetic using finite data types (namely, ints or floats) and idealized arithmetic over the reals or integers. These discrepancies cause the sensitivity of the implemented statistics (i.e., how much one individual's data can affect the result) to be much larger than the sensitivity we expect. Consequently, essentially all differential privacy libraries fail to introduce enough noise to meet the requirements of differential privacy, and we show that this may be exploited in realistic attacks that can extract individual-level information from private query systems. In addition to presenting these vulnerabilities, we also provide a number of solutions, which modify or constrain the way in which the sum is implemented in order to recover the idealized or near-idealized bounds on sensitivity.
- https://arxiv.org/abs/2207.10635

Machine Learning

- Anonymizing Machine Learning Models (2021, IBM)
- There is a known tension between the need to analyze personal data to drive business and privacy concerns. Many data protection regulations, including the EU General Data Protection Regulation (GDPR) and the California Consumer Protection Act (CCPA), set out strict restrictions and obligations on the collection and processing of personal data. Moreover, machine learning models themselves can be used to derive personal information, as demonstrated by recent membership and attribute inference attacks. Anonymized data, however, is exempt from the obligations set out in these regulations. It is therefore desirable to be able to create models that are anonymized, thus also exempting them from those obligations, in addition to providing better protection against attacks. Learning on anonymized data typically results in significant degradation in accuracy. In this work, we propose a method that is able to achieve better model accuracy by using the knowledge encoded within the trained model, and guiding our anonymization process to minimize the impact on the model's accuracy, a process we call accuracy-guided anonymization. We demonstrate that by focusing on the model's accuracy rather than generic information loss measures, our method outperforms state of the art k-anonymity methods in terms of the achieved utility, in particular with high values of k and large numbers of quasi-identifiers. We also demonstrate that our approach has a similar, and sometimes even better ability to prevent membership inference attacks as approaches based on differential privacy, while averting some of their drawbacks such as complexity, performance overhead and model-specific implementations. This makes model-guided anonymization a legitimate substitute for such methods and a practical approach to creating privacy-preserving models.
- https://arxiv.org/abs/2007.13086