Differential privacy in practice

Lessons learned in a real world implementation

Aleatha Parker-Wood, Principal Privacy Engineer, Amazon (all work done prior to Amazon)

ACSAC 2021

Disclaimers

This case study describes work done at my previous employer, Humu, as well as my own thoughts on private system architecture which may or may not reflect past or present systems at Humu. All views expressed here are strictly my own, and do not reflect the views of any employer, past or present.

What is Differential Privacy?

Intuition: Noisy answers protect privacy

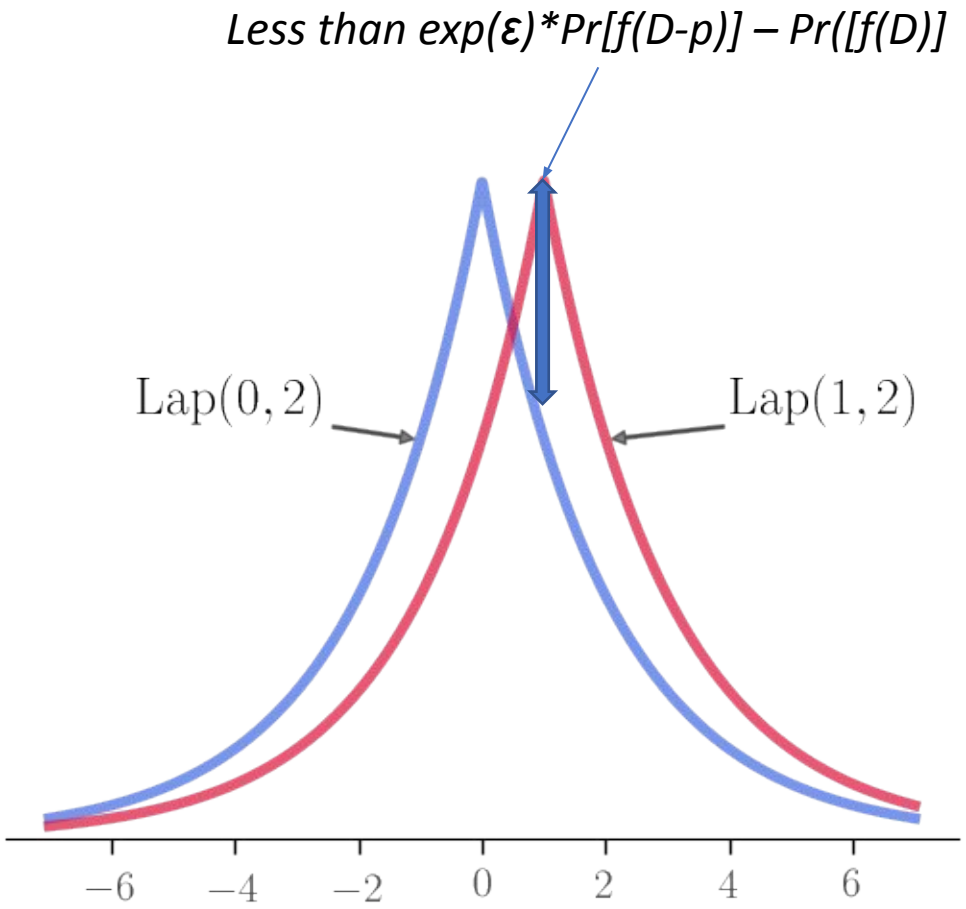
- Airlines want to know they're exceeding the weight limit for a plane
- If they ask everyone on the plane for their weight they'll have unhappy customers
- If they ask everyone for their weight, plus/minus a random number, each person has plausible deniability...
- And yet the calculated total will be close to the true total if you do it right

Statistically rigorous privacy guarantees

- Without loss of generality
- For a single data point p, in dataset D…
- And a function f with image Y..
- F is ε-DP iff:
- Pr[f(D)] <= exp(ε) * Pr[

Statistically rigorous privacy guarantees

- If a person contributes <u>one</u> data point p to any dataset D..
- And we run an differentially private function f over D…
- Regardless of the original data point, and the random number we use to determine the output, p can change the probability of the final statistic by at most a factor of e^ε.



What does that actually mean?

- Differential privacy is a worst case information-theoretic bound
- ε describes the worst case distance between two distributions, [D |f(D)] and [D-p| f(D-p)], describing two alternate worlds where p does/doesn't exist in the data
- A Bayesian attacker's hypothesis about which world is the true one:

$$P(D|f(D)) = \frac{P(f(D)|D)P(D)}{P(f(D)|D)P(D) + P(f(D-p)|D-p)P(D-p)}$$

- Bounding the multiplicative ratios means no matter what random draw we get, and what priors the attacker has, the posterior probability distributions are never no more than e^ ε apart
- Maximum gap between distributions is known as privacy loss, what the attacker learns in worst case

A few more key concepts

Cumulative risk

- Each answer from the data offers the attacker a new improved estimate for the posterior (what world they're in)
- Eventually the truth is revealed for some or all of the data
- A privacy *budget* (i.e. cumulative epsilon) caps the total privacy *loss* (i.e. the posterior knowledge of an attacker)
- Differential privacy is really user-friendly for determining loss
- Privacy loss of query_1 and query_2 is at most the sum of the epsilons

Sensitivity

- To mask one user's input, you need to know how much they can change the outcome
- Average number of cancer patients? One user changes count by 1. $\text{Sensitivity} = 1/n$
- Average weight? One user might contribute between 1-450 kgs. $\text{Sensitivity} = 450/n$

Differential Privacy at Humu

More Disclaimers

- This talk is not intended to be a comprehensive guide to privacy at Humu
- Differential privacy was part of a suite of privacy protections that aren’t relevant here
- Strictly a lessons learned talk for differential privacy system design

Background

- Working with survey data, collected in fixed length epochs (batch mode)
- Survey results can be decomposed in many different ways (by team, by gender, by geography…)
- Statistics are reported out to each company based on their own data

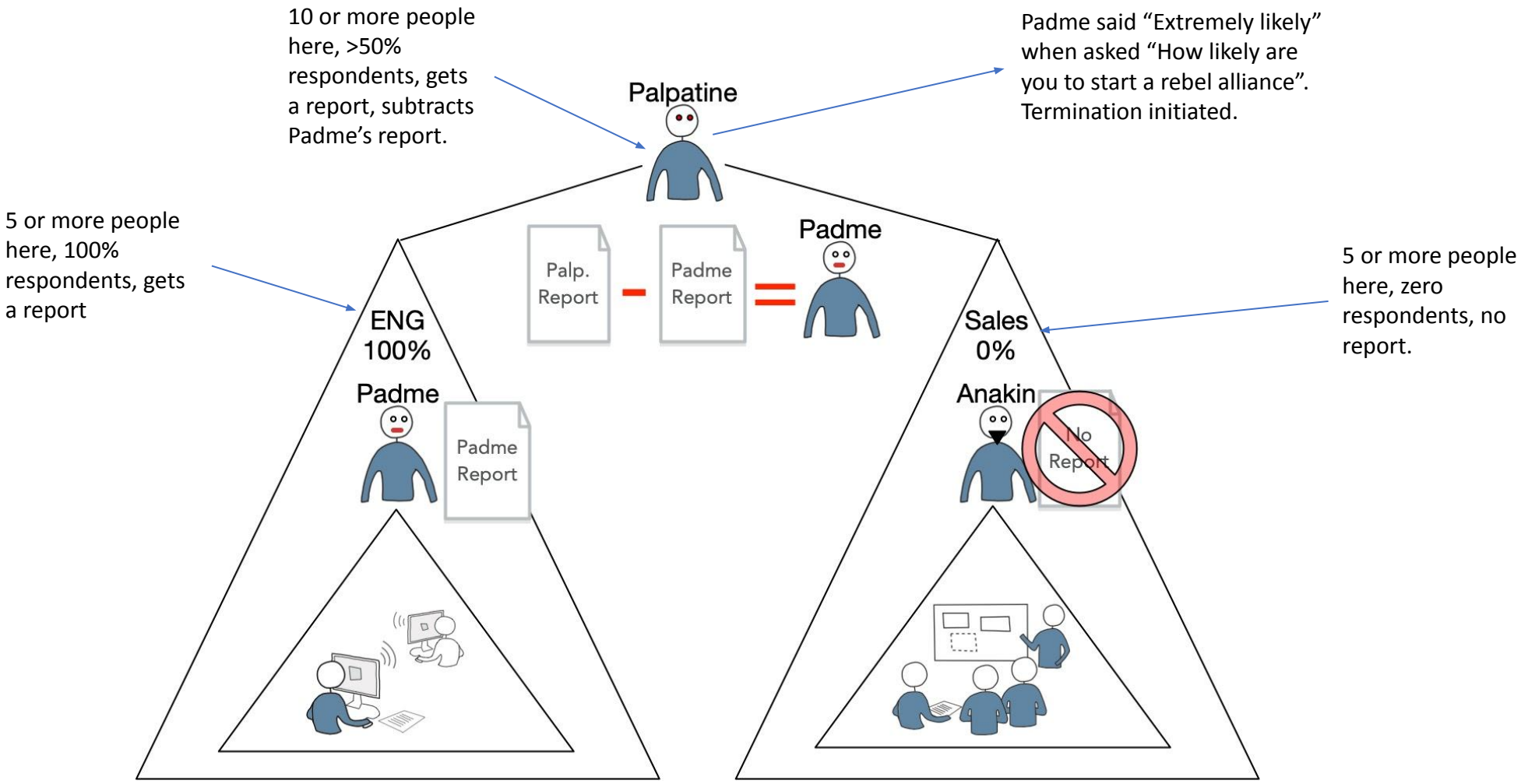
Protect anonymity of survey respondents

- Focus on statistics derived from surveys and metadata
- Privacy should prevent
  - Re-identification
    - “Did Bob reply to this survey? What did he say?”
  - Honest-but-curious people
    - “I wonder who on my team is happiest?”
  - Malicious attackers
    - “I want to find and fire all the people who don’t like it here.”
- When promising anonymity to respondents, the goal is to never allow a survey respondent to be singled out, regardless of what statistics an observer can combine

For example…

- “What is the average survey response rate for this team?” -> Differentially private survey response rates
- ”What percentage of employees feel like communication is very clear/somewhat clear/not clear?” -> Differentially private histograms
- “Does the European office view their prospects for advancement more or less positively on average than the US office at this company?” -> Differentially private averages

K-anon is not enough: Differencing Attacks



Interesting challenges

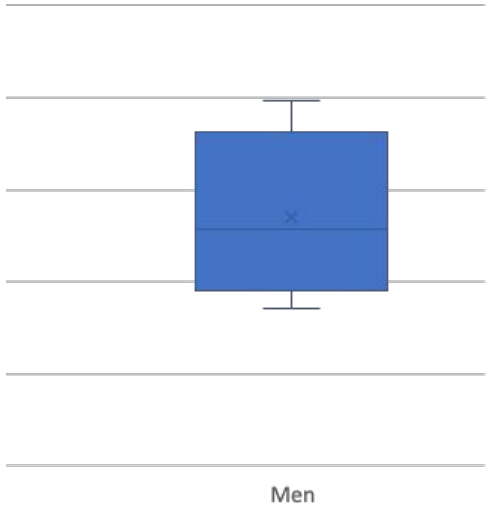
- Data sets are relatively small by ML standards
  - Largest company in the world has 2.2 million employees
  - A regional branch office might only employ 10 people
- Sensitivity can accordingly be quite high
- Active development means exploring new statistics constantly
- Some statistics have academic sensitivity analysis already, others don’t
- Customer facing statistics means getting the usability right

Usability lessons in differential privacy

Thanks to @Kareem_Carr for the meme idea

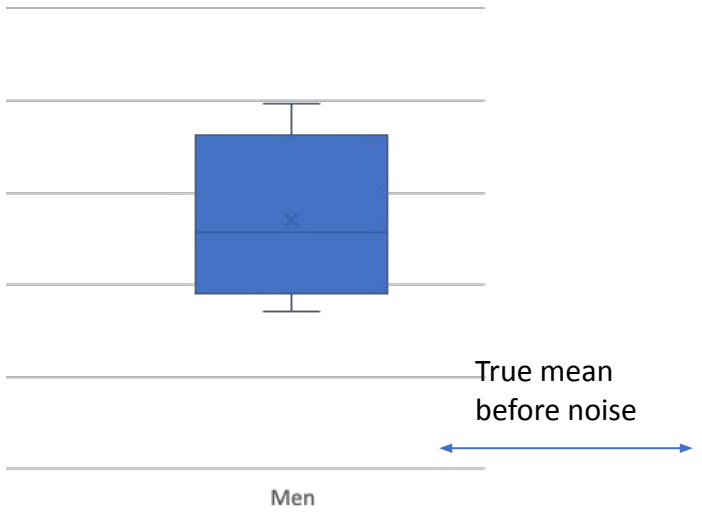
Anchoring, and the psychology of noise

- What if I told you…
- …that women at a certain company are on average 1 point more likely to quit. (+/- 1.5 points?)



Anchoring, and the psychology of noise

- Except they’re not.
- People find numbers and pictures memorable
- Confidence intervals? Not so much
- <u>When you display a noisy number, it’s easy to draw wrong conclusions.</u>



Sensitivity lessons

- Average number of cancer patients? One user changes count by 1
- Average weight? One user might contribute between 1-450 kgs.
- Differentially private torque? ("Who is responsible for this 100 meter diameter space station rotating faster?!") Now it's between 0 and 4,500,000.
- Differential privacy requires you to think differently about the statistics you want, and the order in which you calculate them.
- Not all statistics are useful after differential privacy

Reminder: Too many secrets

- The more statistics are displayed, the more noise needs to be added to maintain the same privacy (see: privacy budget)
- Differential privacy means that data can't be sliced and diced arbitrarily
- Good for privacy, bad when you don't know what questions to ask in advance

Managing your privacy budget

Tracking your privacy budget (easy mode)

- Privacy budget is a property of the dataset, not the query
- If you are only releasing a known set of statistics once.. (e.g. “US Census”)
  - Allocate your privacy budget over all statistics so that it adds up to your total desired privacy loss (evenly, or in a weighted fashion by utility)
  - Calculate differentially private releases for each statistic
  - Store those somewhere, release them to the world
  - Never, ever touch the original data set again

Tracking your privacy budget (running queries, fixed data)

- Maintain epsilon as metadata on the dataset
- You need a way to uniquely identify a dataset snapshot (this is surprisingly hard in many systems)
- Eventual consistency is likely not good enough
  - You need atomic, consistent updates with locks
  - This is going to be a performance bottleneck
- Decide in advance how to allocate your budget
  - Exponentially less accurate answers over time?
  - Evenly distributed noise?
- Cache results for previous queries in a durable store
- Stop answering queries after you run out of budget (this is hard to sell!)

What if you have an evolving data set?

- I did not have to solve for this, but the struggle is real.
- Some options…
  - Decompose your problem such that you only release by epoch (e.g. “we retrain ML models every week”.)
  - Track potential privacy loss by sub-dataset (more on this later)
  - There’s lots of theoretical work in this area for eking out more utility

Hey, is this thing on? DP and QA

- How do you know if your DP is working? Ask it!
- Ok, I got a random number! (Is that good? Bad? I don't know!)
- Two options.
  - Formal verification. (e.g. Wang et al. CCS 2020)
  - Or just ask it enough times to get a statistically valid sample.
- Ok, I asked it once, and then it stopped talking to me!
- That's because you're enforcing your epsilon budget. (Good job.)
- For testing the DP mechanism, you need to turn off epsilon tracking
- Then you need to test the epsilon tracking separately

The truth is out there (Gold master data)

- Once a number is out there, the privacy loss is irrevocable
- If you are doing all the stats at the same time, consider calculating and creating a read-only cache as part of your prod build pipeline
- A push to prod should include tests to confirm epsilon tracking is active

System design concerns

Sensitivity

- Do sensitivity analysis early in product design
- Decide whether you can live with the expected utility
- Some results are so noisy as to be useless, due to small data sets or high sensitivity queries
- Some queries have specialized algorithms for higher utility

Huge tables, tiny queries

- Many queries filter out most of the data before running an operation
- Problem: naïve designs burn budget for the entire table, despite not using most of the data
- Consider partitioning your data based on common queries, and track epsilon individually by partition
- Compute (and cache) differentially private intermediate results and then join
- Use cached results for subsequent sub-tasks in queries

I need to hotfix prod! Now what?

- Can you use the cached result? If so, then you're in the clear.
- If you found a bug in the statistic calculation or dataset…. Now you need to analyze the privacy loss.
- Case 1: The old number has nothing to do with the data. It's just garbage.
  - You're not creating privacy loss, push the fix.
- Case 2: The old number was private and based on the data, but it's the wrong stat.
  - Balance the harm of the wrong number (remember, you're adding noise anyway) against the privacy implications. How wrong is it? If you recalculate, you lose that budget forever.
- Case 3: The old number was not sufficiently private
  - Call in the legal and privacy engineering cavalry. Do not try to fix it without a consult.

Conclusions

DP comes with new design concerns

- Identifiers for data snapshots/subsets
- Potential bottlenecks when updating privacy loss
- Result caches are mandatory and durable
- Testing DP requires statistical knowledge
- Get your statisticians involved early for sensitivity