Introduction to Differential Privacy
CMSC 23200/33250, Winter 2022, Lecture 22

David Cash and Blase Ur
University of Chicago
1. Basic Setting and Ideas for Differential Privacy
2. Local Differential Privacy and Randomized Response
3. (Traditional) Differential Privacy
4. Some Attacks against Differential Privacy Systems
Recalling the Problem Setting

<table>
<thead>
<tr>
<th>name</th>
<th>age</th>
<th>zip</th>
<th>income</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatma</td>
<td>33</td>
<td>60637</td>
<td>25k</td>
</tr>
<tr>
<td>Hong</td>
<td>14</td>
<td>60638</td>
<td>35k</td>
</tr>
<tr>
<td>Roger</td>
<td>21</td>
<td>60637</td>
<td>60k</td>
</tr>
</tbody>
</table>

Individuals → Data Collection → Database → Publish → Analyze
Lessons from Last Time

1. Old methods (e.g. de-identification) provide little protection

2. Principled methods (k-anonymity, l-diversity) also fail often

![Figure 1 Linking to re-identify data](source: L. Sweeney. k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 2002; 557-570.)

![4-Antonymous Inpatient Microdata](source: A. Machanavajjhala et al. l-Diversity: Privacy Beyond k-Anonymity. TKDD 2007.)
Properties of an Ideal Data Privacy Solution

1. Hide all information that may be harmful to individuals.

2. Resist attacks by adversaries with arbitrary background data.

Initial Insights (inspired by Randomized Response)

1. Approximate answers can be (just as) useful.
2. Focus on the *distribution* of what is released, not actual responses.
3. Plausible deniability may be good protection.
Differential Privacy: Main Idea

**Design Philosophy:** Publish data with some random “noise” added. Adding or removing any individual from the data should not change the *distribution* of the output “by too much”.

- If data release does not change much when an individual is included, conclude that they are protected.

<table>
<thead>
<tr>
<th>name</th>
<th>age</th>
<th>zip</th>
<th>income</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatma</td>
<td>33</td>
<td>60637</td>
<td>25k</td>
</tr>
<tr>
<td>Hong</td>
<td>14</td>
<td>60638</td>
<td>35k</td>
</tr>
<tr>
<td>Roger</td>
<td>21</td>
<td>60637</td>
<td>60k</td>
</tr>
</tbody>
</table>

“Noising” procedure:

- If data release does not change much when an individual is included, conclude that they are protected.
Differential Privacy: TODOs

**Design Philosophy**: Publishing data with some random “noise” added. Adding or removing an individual from the data should not change the *distribution* of the output “by too much”.

- How should this noise be chosen? How much noise?
- How should we measure changes in distributions?
- What do these protections mean in practice?
Outline

1. Basic Setting and Ideas for Differential Privacy

2. Local Differential Privacy and Randomized Response

3. (Traditional) Differential Privacy

4. Some Attacks against Differential Privacy Systems
System Architecture for Local Differential Privacy

“Noising” procedure:

Adding or removing one…

---

<table>
<thead>
<tr>
<th>name</th>
<th>Disease 1?</th>
<th>Disease 2?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatma</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Hong</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Roger</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Publish

<table>
<thead>
<tr>
<th>Disease #</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Shouldn’t change the **distribution** of the output too much.
Defining Local Differential Privacy

**Definition**: A randomized algorithm $A$ is \( \varepsilon \)-locally-differentially-private if:
- For every pair of local inputs \( x, x' \)
- For every set \( S \) of possible outputs

It holds that:
\[
\Pr[A(x) \in S] \leq e^{\varepsilon} \cdot \Pr[A(x') \in S]
\]

- \( e \approx 2.71 \) is Euler’s number. Could just use \( \varepsilon \) instead of \( e^{\varepsilon} \)
- Definition is symmetric in \( x, x' \).
- The event “\( A(x) \in S \)” can represent any observation as the set \( S \) changes. (“Average was in some range” or “Even number of people had disease”).
- Smaller \( \varepsilon \) means better privacy. \( \varepsilon=0 \) means distributions are same.
Fix any local inputs $x, x'$, running $A(x)$ and $A(x')$ will induce two distributions that we hope are close. We know:

$$\Pr[A(x) \in S] \leq e^\varepsilon \cdot \Pr[A(x') \in S]$$
Randomized Response is Locally-DP

**Definition of** $A_{rr}$: Takes an input $x \in \{Y, N\}$.
- With probability 0.5, $A_{rr}(x) = x$
- With probability 0.5, $A_{rr}(x)$ outputs $Y$ or $N$ uniformly at random.

**Claim:** $A_{rr}$ is $\varepsilon$-locally-DP for $\varepsilon = \ln 3 \approx 1.10$.

**Proof:** Must show for all $S \subseteq \{Y, N\}$ and all $x, x' \in \{Y, N\}$,

$$\Pr[A_{rr}(x) \in S] \leq e^{\varepsilon} \cdot \Pr[A_{rr}(x') \in S].$$

Only need to consider $x = Y, x' = N$ and $S = \{Y\}$ or $S = \{N\}$:

- $\Pr[A_{rr}(Y) = Y] = 0.5 + 0.5 \cdot 0.5 = 0.75$
- $\Pr[A_{rr}(N) = Y] = 0.5 \cdot 0.5 = 0.25$ (others are similar)

Checking cases, $\Pr[A_{rr}(x) \in S] \leq 3 \cdot \Pr[A_{rr}(x') \in S]$ always. In other words: $A_{rr}$ is $\varepsilon$-locally-DP for $\varepsilon = \ln 3$. 
Deployed Local DP: Apple iPhone Data Collection

- Per-user table is supposedly not actually stored
- Also collecting: Power usage, text slang (!), …
Outline

1. Basic Setting and Ideas for Differential Privacy
2. Local Differential Privacy and Randomized Response
3. (Traditional) Differential Privacy
4. Some Attacks against Differential Privacy
System Architecture: (Traditional) Differential Privacy

“Noising” procedure:

Adding or removing one…

<table>
<thead>
<tr>
<th>name</th>
<th>Disease 1?</th>
<th>Disease 2?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatma</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Hong</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Roger</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Publish

<table>
<thead>
<tr>
<th>Disease #</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Shouldn’t change the distribution of the output too much.
System Architecture: (Traditional) Differential Privacy

“Noising” procedure:

Adding or removing one...

<table>
<thead>
<tr>
<th>name</th>
<th>Disease 1?</th>
<th>Disease 2?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatma</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Hong</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Roger</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Individuals

Database

Query $f(D)$

$\text{Shouldn’t change the } \text{distribution of the output too much.}$

• Noised version of $f(D)$ is usually denoted $\mathcal{M}(D)$
Simplifying the Problem: Abstract “Databases”

<table>
<thead>
<tr>
<th>Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

- Data $D$ is a table of counts.
- Query $f$ can be an arbitrary function of $D$.

**Definition:** Datasets $D$, $D'$ are neighboring if they are exactly the same, except they differ by exactly 1 in a single count.

Example:

<table>
<thead>
<tr>
<th>Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>
Defining Differential Privacy

**Definition:** A randomized algorithm $\mathcal{M}$ is $\epsilon$-differentially-private if:
- For every pair of neighboring tables $D, D'$
- For every set $S$ of possible outputs

It holds that:
$$\Pr[\mathcal{M}(D) \in S] \leq e^\epsilon \cdot \Pr[\mathcal{M}(D') \in S]$$

• Goal: Add as little noise as possible while respecting definition
Calibrating Noise: Sensitivity of a Query

**Definition**: The sensitivity of a function $f$, denoted $\Delta f$, is defined to be

$$\Delta f = \max_{D, D'} |f(D) - f(D')|$$

where the maximum is taken over neighboring pairs of tables $D, D'$.

- Adding someone to $D$ can change $f(D)$ by at most $\Delta f$.
- Plan: Add more noise when $\Delta f$ is large, to hide effect of individual.
- Won’t need to worry about any other property of $f$. 

Sensitivity of a Query: Examples

Database $D$:

<table>
<thead>
<tr>
<th>Disease #</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>94</td>
</tr>
<tr>
<td>5</td>
<td>77</td>
</tr>
</tbody>
</table>

Query $f(D)$: Output number with disease #1

Question: What is $\Delta f$?

Database $D$:

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatma</td>
<td>21</td>
</tr>
<tr>
<td>Hong</td>
<td>20</td>
</tr>
<tr>
<td>Ron</td>
<td>35</td>
</tr>
<tr>
<td>Oren</td>
<td>25</td>
</tr>
</tbody>
</table>

Query $f(D)$: Output number of people over 21

Question: What is $\Delta f$?

Database $D$:

<table>
<thead>
<tr>
<th>Name</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatma</td>
<td>35k</td>
</tr>
<tr>
<td>Hong</td>
<td>20k</td>
</tr>
<tr>
<td>Ron</td>
<td>100k</td>
</tr>
<tr>
<td>Oren</td>
<td>50k</td>
</tr>
</tbody>
</table>

Query $f(D)$: Average income

Question: What is $\Delta f$?
**Definition:** The *Laplace distribution (centered at zero)* with scale $b$ is defined to have probability density function

$$\frac{1}{2b} e^{-|x|/b}.$$

- Larger scale $\Rightarrow$ More variance

---

b=1 (blue) and b=2 (red)
The Laplace Mechanism

**Definition:** The Laplace Mechanism $\mathcal{M}$ for a query $f$ with privacy parameter $\epsilon$ is defined to be

$$\mathcal{M}(D) = f(D) + \text{Laplace}(\Delta f/\epsilon).$$

- Larger $\Delta f$ $\Rightarrow$ Larger scale $\Rightarrow$ More variance $\Rightarrow$ Less utility
- Smaller $\epsilon$ $\Rightarrow$ Larger scale $\Rightarrow$ More variance $\Rightarrow$ Less utility
- Can show: This is “optimal” distribution amongst $\epsilon$-DP mechanisms.

**Claim:** $\mathcal{M}$ is $\epsilon$-DP.
Flaws in DP Systems (Assignment 8)

<table>
<thead>
<tr>
<th>name</th>
<th>Disease 1?</th>
<th>Disease 2?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatma</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Hong</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Roger</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

- If adversary can repeat query many times…
- Average of results will be true answer.
- In practice, systems must manage a “privacy budget”
import numpy.random

def laplace_mechanism(val, sensitivity, epsilon):
    noise = numpy.random.laplace(0.0, scale=sensitivity/epsilon)
    return val + noise

- Numeric variables above are floating point. Not all numbers are representable.

On Significance of the Least Significant Bits For Differential Privacy

Ilya Mironov
Floating Point and Laplace Mechanism (Assignment 8)

**Attack setting:**
- Adversary knows $f(D)$ is either 0 or 1
- Adversary gets to see $\mathcal{M}(D) = f(D) + \text{Laplace}(\Delta f/\varepsilon)$
- Adversary tries to guess $f(D)$
- Adversary should do no better at guessing $f(D)$ than is allowed by $\varepsilon$-DP

**Key insight:** Most Laplace samplers do not output every possible floating point. Some numbers will never be output.

Representable floating point numbers:

\[ \begin{array}{cccccccccccccccc}
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array} \]

- Orange square: Sampler outputs with non-zero probability
- Gray square: Sampler will never output
Floating Point and Laplace Mechanism (Assignment 8)

\[ \mathcal{M}(D) = 0 + \text{Laplace}(\Delta f / \varepsilon) \]

or

\[ \mathcal{M}(D) = 1 + \text{Laplace}(\Delta f / \varepsilon) \]

→ Sampler outputs with non-zero probability

→ Sampler will never output

Possible outputs when \( f(D) = 0 \) (i.e. \( \mathcal{M}(D) = \text{Laplace}(\Delta f / \varepsilon) \)):

```
... 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...`
```

Possible outputs when \( f(D) = 1 \) (i.e. \( \mathcal{M}(D) = 1 + \text{Laplace}(\Delta f / \varepsilon) \)):

```
... 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...`
```
Floating Point and Laplace Mechanism (Assignment 8)

\[ \mathcal{M}(D) = 0 + \text{Laplace}(\Delta f/\varepsilon) \]

or

\[ \mathcal{M}(D) = 1 + \text{Laplace}(\Delta f/\varepsilon) \]

⇒ Sampler outputs with non-zero probability

⇒ Sampler will never output

Possible outputs when \( f(D) = 0 \) (i.e. \( \mathcal{M}(D) = \text{Laplace}(\Delta f/\varepsilon) \)):

Possible outputs when \( f(D) = 1 \) (i.e. \( \mathcal{M}(D) = 1 + \text{Laplace}(\Delta f/\varepsilon) \)):

⇒ “Smoking gun” samples that would only be output in one case.
The End