- Exam
- Scribes?
- Lecture recording

Centralized model:

\[ \sum x_i + \text{Lap}(\frac{\gamma}{e}) \]

- \( \epsilon \)-differential private
- Noise \( O(\frac{\gamma}{e}) \)

Local model: Local Laplace mechanism. Each party sends \( x'_i = x_i + \text{Lap}(\frac{\gamma}{e}) \) and curator releases \( \sum x'_i \)

- \( \epsilon \)-differential private
- Noise \( O(\frac{\gamma}{\epsilon}) \) - optimal

Shuffle model
I.e., to generate a noisy histogram:

- Each party applies randomized response
  - With prob. \(1 - \delta\), \(x'_i = x_i\)
  - With prob. \(\delta\), \(x'_i \leftarrow \{1, \ldots, k\}\)
- Each party sends \(x'_i\) to anonymous bulletin board
- Curator gets \(\{x'_1, \ldots, x'_k\}\) and generates histogram from that

In expectation, \(\delta\)-fraction of the parties replace their inputs by random value.

Consider changing input of \(P_i\)

Case 1: \(P_i\) sends uniform value in \(\{1, \ldots, k\}\)
- View is independent of \(P_i\)'s input

Case 2: \(P_i\) sends its input (either \(x_i\) or \(x'_i\))
- Still in private but not in shuffle model
- By w.p. some probability, other parties send \(x_i/x'_i\) due to randomized response

Fix any view of the curator — including bit-vector indicating which parties sent random values.

Let \(V\) denote the values sent by \(P_i\) — the values sent by the parties sending random values.

\[
\Pr \left[ M(1, \ldots, x_n) = V \right] = \frac{1!}{n_{-1}! n_2 \cdots n_k} \cdot \Pr \left[ \text{IB1 parties send random values} \right]
\]

\[
\Pr \left[ M(2, x_2, \ldots, x_n) = V \right] = \frac{1!}{n_1! n_{-1} \cdots n_k} \cdot \Pr \left[ \text{IB1 parties send random values} \right]
\]

\[
\left( \frac{1!}{n_{-1}! n_2 \cdots n_k} \right) = \left( \frac{1!}{n_{-1}!} \right) \cdot \left( \frac{1!}{n_2} \right) \cdot \left( \frac{1!}{n_3} \right) \cdots
\]

\[
\left( \frac{1!}{n_1!} \right) = \left( \frac{1!}{n_{-1}!} \right) \cdot \left( \frac{1!}{n_2} \right) \cdot \left( \frac{1!}{n_3} \right) \cdots
\]
\[ P_c(M(1, x_1, \ldots, x_n) = u) = \frac{(\frac{n!}{(n-1)!}) \cdot (\left( \frac{n! - n, u-1}{n^2} \right)}{P_c(M(2, x_2, \ldots, x_n) = u)} = \frac{(\frac{n!}{(n-1)!}) \cdot (\left( \frac{n! - n, u-1}{n^2} \right)}{P_c(M(2, x_2, \ldots, x_n) = u)} = \frac{n! \cdot (n, u-1)!}{n^2 \cdot (n-1)!} = \frac{n!}{n^2} \]

Application to summation
- Use this protocol to generate a histogram
- Compute the sum of the contributed values

\[ \Rightarrow \text{noise } O(n^{1/2}) \]

Assuming \( \xi = O\left(\log \left(\frac{m}{\log m}\right)\right) \)

Amplification result: take any \( R \) that is \( \xi_0 \)-LDP and run it through the shuffle mechanism. The result is \( O\left(\min \{\xi, 1\} \cdot \left(\frac{\log (\log m)}{m^{1/2}}\right)^{\xi}\right) \)-DP

---

Multi-message shuffle model

\[
\begin{align*}
\text{C} & \rightarrow ?x_1^i, x_2^i, \ldots, x_n^i \rightarrow \text{Ann} \rightarrow r_1, \ldots, r_n \rightarrow P_1, \ldots, P_n \\
\text{x}_1^i, \ldots, x_n^i & \rightarrow r_1, \ldots, r_n \rightarrow P_1, \ldots, P_n
\end{align*}
\]

Single-message shuffle model

\[
\begin{align*}
\text{C} & \rightarrow ?(x_1^1, x_2^1, \ldots, x_n^1) \rightarrow \text{Ann} \rightarrow r_1, \ldots, r_n \rightarrow P_1, \ldots, P_n \\
\text{x}_1^1, \ldots, x_n^1 & \rightarrow r_1, \ldots, r_n \rightarrow P_1, \ldots, P_n
\end{align*}
\]
Parallel multi-message shuffling

\[ \{ x_i, \ldots, x_n \} \rightarrow \{ x'_i, \ldots, x'_n \} \]

\[ \text{anon}_i \quad - \quad \text{anon}_n \]

\[ x_i' \quad \text{anon}_i \quad x_n' \]

\[ r_i \quad \text{anon}_i \quad r_n \]

Exact summation using anonymization layer

- Want to learn the sum exactly, without revealing parties inputs

View \((\bar{x})\) vs View \((\bar{x}')\) For any \(x, x'\) with the same sum

Protocol: Each party \(p_i\) with input \(x_i\), chooses \(m\) values \(x_i', \ldots, x_i'^m\)
uniformly subject to \(\sum x_i'^{\ell} = x_i\) and \(\ell\)

Send \(x_i', \ldots, x_i'^m\) to the anon. service

Curator gets \(\{ x_i', \ldots, x_i'^m, \ldots, x_n', \ldots, x_n'^m \} \) and outputs the sum

To achieve differential privacy, apply this protocol to \(x_i + \text{noise}\);

Final result: \(\sum x_i + \sum \text{noise}_i\) \(\in \mathcal{O}(\epsilon^2)\)

Interactive local model

General protocols