- Scribes?
- Lecture recording

\[
\sum_{i=1}^{n} x_i
\]

Want to compute \( f(x_1, \ldots, x_n) = \sum x_i \)

- Instead compute approximation \( \tilde{f}(\tilde{x}) = R(x_1^* \cdot \text{Lap}(\frac{\epsilon}{n})) \)
- To avoid a central authority, use MPC to compute \( f \)

\[\Rightarrow \text{prob} \tilde{f} \text{ that computes a dp. approx to } f\]

Infor-theoretic diff. privacy of \( TT \):

For any set of \( T \) parties and any neighboring inputs \( \tilde{x}, \tilde{x}' \) (that are equal for the \( T \) corrupted), \( \forall \) any set of views \( V \)

\[
P_c \left( \text{View}_T^i(\tilde{x}) \in V \right) \leq e^\epsilon \cdot P_c \left( \text{View}_T^i(\tilde{x}') \in V \right)
\]

Computational version of diff. privacy of \( TT \)

For all efficient distinguishers \( D \):

\[
P_c \left( D(\text{View}_T^i(\tilde{x})) = 1 \right) \leq e^\epsilon \cdot P_c \left( D(\text{View}_T^i(\tilde{x}')) = 1 \right) + \delta(n)
\]

Centralized protocol for summation: \( \sum, x_i \cdot \text{Lap}(\frac{\epsilon}{n}) \)

Local protocol for summation: \( \sum, (x_i + \text{Lap}(\frac{\epsilon}{n})) \)

(un) Computationally dp. protocol for summation:
- Parties set up a threshold homomorphic encryption scheme
  - Public key $pk$
  - Given $Enc_{pk}(x_i)$, $Enc_{pk}(x_j) \Rightarrow Enc_{pk}(x_i \cdot x_j)$
  - Threshold: every party holds a share $sk_i$ of secret key $sk$
- Every party sets $x'_i = x_i + \text{noise}$
- Publish $Enc_{pk}(x'_i) <$
- Parties compose $Enc_{pk}(\sum_i x'_i)$
- Parties collectively decrypt to get $\sum_i x'_i$

$\Rightarrow$ noise per party can be much lower than in the CDP

Here: use MPC to compute a differentially private functionality $f$

Another possibility: Could use a $(\varepsilon, \delta)$-DP MPC protocol to compute $f$

3rd possibility: use a $(\varepsilon, \delta)$-DP MPC protocol to compute $f$

Semi-honest:

Protocol $\varepsilon$-differentially private computes $f$: $f$:
- $\Pi$ secretly computes $f^L$
- $L$ is $\varepsilon$-d.p.
\[
\begin{align*}
\mathbf{F}_1 (\mathbf{x}) & \quad \xrightarrow{\text{Server 1}} \quad \xrightarrow{\text{Server 2}} \quad \mathbf{F}_2 (\mathbf{x}) \\
\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_n & \quad \xrightarrow{\mathbf{F}_1} \quad \xrightarrow{\mathbf{F}_2} \\
\mathbf{y}_1 \quad \cdots \quad \mathbf{y}_n & \quad = \quad \mathbf{F} (\mathbf{x})
\end{align*}
\]
diff. private computation, malicious case

\[ \mathcal{L} = \mathcal{F} (\mathcal{L}_1, \mathcal{L}_2) \]

Protocol \( \Pi \) is \( \epsilon \)-d.p. if
- \( \Pi \) securely realizes \( \mathcal{F} \)
- Every \( (\mathcal{L}_1, \mathcal{L}_2) \in \mathcal{L} \) is \( \epsilon \)-differentially private

\[ P_1(x_1, \ldots, x_n) \quad \quad \quad \quad P_2(y_1, \ldots, y_n) \]

\[ d, d', x_1, x_2, x_3 \leftarrow Y_1, Y_2, Y_3 \]

\[ y_1, y_2, d_1, d_2 \]

\[ y_1, y_2, y_3 \]

\[ y_3 \]