Pan-Private Uniformity Testing

Matthew Joseph

Kareem Amin  Jieming Mao
Models of Differential Privacy

- Local
- Pan
- Central
Models of Differential Privacy

User Trust in Algorithm Operator

- Local
- Pan
- Central

Utility
Models of Differential Privacy

User Trust in Algorithm Operator

Utility

Local

Pan

Central

(trust today, not tomorrow)

(trust today, and tomorrow)

(no trust)
This Talk

Models of Differential Privacy

(trust today, and tomorrow)

Central

( trust today, not tomorrow)

Pan

(no trust)

Local

Utility

User Trust in Algorithm Operator
Outline

1. Local Privacy Basics
2. Pan-Privacy Basics
3. Result 1: Connecting Local and Pan-Privacy
4. Result 2: Pan-Private Uniformity Testing
Outline

1. Local Privacy Basics
2. Pan-Privacy Basics
3. Result 1: Connecting Local and Pan-Privacy
4. Result 2: Pan-Private Uniformity Testing
Local DP Learning From Data

Data

Noise

Learning

Output

Local Differential Privacy
Local DP in Words

Distributed database, users keep their data

Protocol $A$ learns about the data through public communication with users

Users send responses through randomizers $R$, differentially private functions of one datum
Types of LDP Interactivity

Definition: Protocol $A$ is **sequentially interactive** [DJW13] if all users speak once (possibly in multiple rounds).
Local DP in Math

Definition: Sequentially interactive protocol $A$ is $(\varepsilon, \delta)$-locally differentially private (LDP) if all randomizers are $(\varepsilon, \delta)$-randomizers.

$$(P[R(x) \text{ in } Y] \leq e^{\varepsilon}P[R(x') \text{ in } Y] + \delta)$$
Outline

1. Local Privacy Basics

2. Pan-Privacy Basics

3. Result 1: Connecting Local and Pan-Privacy

4. Result 2: Pan-Private Uniformity Testing
Local DP Learning From Data

Data → Noise → Learning → Output

Pan-Privacy
Pan-Private [DNPRY10] Learning From Data

Data → Learning → Noisy State → Output

Pan-Privacy
Pan-Privacy in Words

Data arrives in a stream, one element at a time

Algorithm $A$ sees element, updates internal state, continues

Adversary sees (any) one internal state and final output, and this view must be a differentially private function of the stream

See data (easier than local), private intermediary state (harder than central)
Pan-Privacy in Math

Definition: Streams $S$ and $S'$ are neighbors if they differ in at most one stream element. Protocol $A$ is $(\varepsilon, \delta)$-pan private against one intrusion if, for all neighboring $S$ and $S'$, times $t$, internal state subsets $I$, and output subsets $O$,

$$P[I(S \leq t) \text{ in } I, O(S \leq t \circ S > t) \text{ in } O] \leq e^{\varepsilon}P[I(S' \leq t) \text{ in } I, O(S' \leq t' \circ S' > t') \text{ in } O] + \delta.$$
Why Pan-Privacy?
Why Pan-Privacy?

Most useful when user trusts operator today, but wants to “future-proof” their data

Examples: worried about government subpoena or operator ownership changes

If user trusts the operator today, privacy of intermediate state protects against future intrusions
Why Pan-Privacy?

User Trust in Algorithm Operator

Utility

Local

Pan

Central

Pan-Privacy
Q: Does the one-intrusion assumption matter?
Q: Does the one-intrusion assumption matter?

A: Yes
Outline

1. Local Privacy Basics
2. Pan-Privacy Basics
3. Result 1: Connecting Local and Pan-Privacy
4. Result 2: Pan-Private Uniformity Testing
Result 1: Pan- vs. Local

Theorem: Any algorithm $A_P$ that is $\varepsilon$-pan-private against two intrusions can be converted into an identical sequentially interactive $\varepsilon$-LDP protocol $A_S$, and vice-versa.
Result 1: Pan- vs. Local

Theorem: Any algorithm $A_P$ that is $\epsilon$-pan-private against two intrusions can be converted into an identical sequentially interactive $\epsilon$-LDP protocol $A_S$, and vice-versa.

So if you need privacy against multiple intrusions, may as well use local privacy.
Result 1: Pan- vs. Local

Proof Sketch
Local to pan: run a local protocol and maintain transcript as internal state.
Result 1: Pan- vs. Local

Proof Sketch
Local to pan: run a local protocol and maintain transcript as internal state.

Pan to local: adversary sees two internal states, can “diff” them. So must randomize whenever update internal state. Randomize every state ≈ sequential interactivity.
Q: Is single-intrusion pan-privacy meaningful?

Result 1: Pan- vs. local
Q: Is single-intrusion pan-privacy meaningful?

A: We suggest yes
Why Single-Intrusion Pan-Privacy?

Single-intrusion pan-privacy suffers when a user contributes data between intrusions ("diff" attack)

Users most worried about giving data to an operator that’s already compromised

For users who trust operator today, single-intrusion pan-privacy is useful (and more private than central)
Outline

1. Local Privacy Basics
2. Pan-Privacy Basics
3. Result 1: Connecting Local and Pan-Privacy
4. Result 2: Pan-Private Uniformity Testing
Result 2: Pan-Private Uniformity Testing

Uniformity testing: algorithm receives samples from unknown distribution $p$ over $[k]$ and must distinguish $p = U_k$ from

$$\|p - U_k\|_{TV} \geq \alpha \text{ w.p. } \geq \frac{2}{3}$$
## Result 2: Pan-Uniformity Testing

<table>
<thead>
<tr>
<th></th>
<th>Previous Work</th>
<th>This Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Privacy</td>
<td>$\Theta(k^{1/2})$ [CDVV14]</td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$-DP</td>
<td>$\Theta(k^{1/2})$ [ASZ18]</td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$-Pan Privacy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SI $\varepsilon$-LDP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NI $\varepsilon$-LDP</td>
<td>$\Theta(k)$ [ACFT19]</td>
<td></td>
</tr>
</tbody>
</table>
## Result 2: Pan- Uniformity Testing

<table>
<thead>
<tr>
<th>Without Privacy</th>
<th>Previous Work</th>
<th>This Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε-DP</td>
<td>$\Theta(k^{1/2})$ [CDVV14]</td>
<td>$\Theta(k^{1/2})$ [ASZ18]</td>
</tr>
<tr>
<td>ε-Pan Privacy</td>
<td>$\Theta(k^{2/3})$</td>
<td>$\Theta(k^{2/3})$</td>
</tr>
<tr>
<td>SI ε-LDP</td>
<td>$\Theta(k)$ [ACFT19]</td>
<td>$\Theta(k)$ [ACFT19]</td>
</tr>
</tbody>
</table>
**Result 2: Pan-Uniformity Testing**

<table>
<thead>
<tr>
<th></th>
<th>Previous Work</th>
<th>This Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Privacy</td>
<td>$\Theta(k^{1/2})$ [CDVV14]</td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$-DP</td>
<td>$\Theta(k^{1/2})$ [ASZ18]</td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$-Pan Privacy</td>
<td></td>
<td>$\Theta(k^{2/3})$</td>
</tr>
<tr>
<td>SI $\varepsilon$-LDP</td>
<td>$\Theta(k)$ [ACFT19]</td>
<td></td>
</tr>
<tr>
<td>NI $\varepsilon$-LDP</td>
<td>$\Theta(k)$</td>
<td></td>
</tr>
</tbody>
</table>
Result 2: Pan-Uniformity Testing

Theorem: $\epsilon$-pan-private uniformity testing has sample complexity

$$\Omega \left( \frac{k^{2/3}}{\alpha^{4/3} \epsilon^{2/3}} + \frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha \epsilon} \right)$$
Result 2: Pan- Uniformity Testing

Upper Bound Sketch
Result 2: Pan-Uniformity Testing

Upper Bound Sketch
Key idea: split difference between central and local approaches
Result 2: Pan- Uniformity Testing

Upper Bound Sketch
Central [CDK17, ADR18, ASZ18]: uses “fine” statistic

Looks at sample counts for all $k$ elements and measures departure from expected count under uniform distribution

Need to add noise to each count to be pan-private
Result 2: Pan- Uniformity Testing

Upper Bound Sketch
Central [CDK17, ADR18, ASZ18]: uses “fine” statistic

Looks at sample counts for all $k$ elements and measures departure from expected count under uniform distribution

Need to add noise to each count to be pan-private. Can get pan- $O(k^{3/4})$ like this...but can we do better?
Result 2: Pan- Uniformity Testing

Upper Bound Sketch
Central [CDK17, ADR18, ASZ18]: uses “fine” statistic

Maybe pan- should use a coarser statistic?
Result 2: Pan-Uniformity Testing

Upper Bound Sketch
Local [ACFT19]: uses coarse statistic

Randomly halves domain, now uniformity testing over [2]
Result 2: Pan- Uniformity Testing

Upper Bound Sketch
Local [ACFT19]:

Result 2: Pan-Private Uniformity Testing
Result 2: Pan-Uniformity Testing

Upper Bound Sketch
Local [ACFT19]:

\[ S1 = \{2,3,6\} \quad \text{and} \quad S2 = \{1,4,5\} \]
Result 2: Pan- Uniformity Testing

Upper Bound Sketch
Local [ACFT19]: uses coarse statistic

Small response domain: good for local!

But sacrifices a lot of testing distance: $\alpha$ to $\alpha/k^{1/2}$ ... so end up using $O(k)$ samples
Result 2: Pan- Uniformity Testing

Upper Bound Sketch
Local [ACFT19]: uses coarse statistic

Maybe pan- should maintain a finer statistic?
Result 2: Pan- Uniformity Testing

Upper Bound Sketch
Pan: coarser than central, finer than local

Randomly partition domain into $n$ equal-size groups, now uniformity testing over $[n]$
Result 2: Pan- Uniformity Testing

Upper Bound Sketch
Pan:
Result 2: Pan- Uniformity Testing

Upper Bound Sketch
Pan:

\[ S_1 = \{1,3\} \quad S_2 = \{5,6\} \quad S_3 = \{2,4\} \]
Result 2: Pan-Uniformity Testing

**Upper Bound Sketch**
Pan: coarser than central, finer than local

Testing distance change is $\alpha$ to $\alpha(n/k)^{1/2}$

Pick $n = \Theta(k^{2/3}\varepsilon^{4/3}/\alpha^{4/3})$ to trade off coarse (not too much noise per bin) and fine (preserve testing distance)
Theorem: $\varepsilon$-pan-private uniformity testing has sample complexity

\[ O \left( \frac{k^{2/3}}{\alpha^{4/3} \varepsilon^{2/3}} + \frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha \varepsilon} \right) \]
Result 2: Pan- Uniformity Testing

Lower Bound Sketch
Adapts information theory lower bound from [DGKR19] for uniformity testing under memory restrictions
Result 2: Pan- Uniformity Testing

Lower Bound Sketch
Adapts information theory lower bound from [DGKR19] for uniformity testing under memory restrictions

Main contribution: replacing memory restriction with privacy restriction
Result 2: Pan-Uniformity Testing

Theorem: $\varepsilon$-pan-private uniformity testing has sample complexity

$$O\left(\frac{k^{2/3}}{\alpha^{4/3} \varepsilon^{2/3}} + \frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha \varepsilon}\right)$$

$$\Omega\left(\frac{k^{2/3}}{\alpha^{4/3} \varepsilon^{2/3}} + \frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha \sqrt{\varepsilon}} + \frac{1}{\alpha \varepsilon}\right)$$
Takeaways

- Pan-privacy is appropriate when user trusts algorithm operator today but maybe not tomorrow
Takeaways

● Pan-privacy is appropriate when user trusts algorithm operator today but maybe not tomorrow

● Pan-privacy against more than one intrusion is equivalent to sequentially interactive local privacy
Takeaways

- Pan-privacy is appropriate when user trusts algorithm operator today but maybe not tomorrow
- Pan-privacy against more than one intrusion is equivalent to sequentially interactive local privacy
- Pan-privacy against a single intrusion trades off both utility and privacy between central and local models
  - $\Theta(k^{1/2})$, $\Theta(k^{2/3})$, and $\Theta(k)$ uniformity testing bounds
Open Questions

- Uniformity testing:
  - close gap between pan upper and lower bounds
  - fully interactive locally private lower bound?
Open Questions

- Uniformity testing:
  - close gap between pan upper and lower bounds
  - fully interactive locally private lower bound?

- What about $(\varepsilon, \delta)$-pan-privacy?
Open Questions

- Uniformity testing:
  - close gap between pan upper and lower bounds
  - fully interactive locally private lower bound?

- What about $(\varepsilon, \delta)$-pan-privacy?

- How powerful is pan-privacy in general?
References