Exponential Separations In Local Differential Privacy

Matthew Joseph

Jieming Mao

Aaron Roth
Problem
Problem

Surgeon General Jerome Adams
“How many Americans have used a schedule-I drug?"
Problem

“How many Americans have used a schedule-I drug?

People are probably reluctant to tell the federal government honestly…

Surgeon General Jerome Adams
“How many Americans have used a schedule-I drug?

People are probably reluctant to tell the federal government honestly...

...so how can I get an accurate answer while guaranteeing plausible deniability for everyone?”
Solution: Randomized Response [W65]
Solution: Randomized Response [W65]
Solution: Randomized Response [W65]
Solution: Randomized Response [W65]

\[ O\left(\sqrt{\#\text{ responses}}\right) \text{ accuracy} \]
Solution: Randomized Response [W65]

$O(\sqrt{\# \text{ responses}})$ accuracy

Randomness $\rightarrow$ plausible deniability
“Sounds good! Let’s do that.”

Surgeon General
Jerome Adams
“Or maybe we can do better if we ask many questions?

First ask person one q, then use the answer to ask person second q, and so on.”

“Sounds good! Let’s do that.”
“Or maybe we can do better if we ask many questions? First ask person one q, then use the answer to ask person second q, and so on.”

“Sounds cumbersome! We need **proof** that the extra effort is worth it first.”

“Sounds good! Let’s do that.”
This Talk
Prove adaptive questioning with plausible deniability is worth it.
This Talk

Prove adaptive questioning with plausible deniability is worth it.

≈

Construct problem where we can prove fully interactive locally differentially private protocols get much better sample complexity than sequentially interactive ones.
Outline

1. Preliminaries

2. Tool: LDP \approx Noisy Communication

3. Application: Exponential Separation
Outline

1. Preliminaries

2. Tool: LDP $\approx$ Noisy Communication

3. Application: Exponential Separation
Local Differential Privacy (LDP) [DMNS06]

Each user has their own private datum

*Protocol* $\mathcal{A}$ learns about the data through public communication with users

Users send messages through *randomizers* $\mathcal{R}$

Randomness ensures privacy
LDP in Math

Definition: Protocol $A$ is $(\varepsilon, \delta)$-locally differentially private (LDP) if the transcript of communications it generates is an $(\varepsilon, \delta)$-DP function of the user data.

For neighboring distributed databases $X$ and $X'$,

$$
P[T(X) \text{ in } Y] \leq e^\varepsilon P[T(X') \text{ in } Y] + \delta$$
LDP: Pros and Cons

Pros:

✓ Data never leaves user device, only DP outputs

✓ Don’t have to store any private data
LDP: Pros and Cons

Pros:
✅ Data never leaves user device, only DP outputs
✅ Don’t have to store any private data

Cons:
✗ More noise → worse utility
✗ Don’t get to store any private data
Types of LDP Interactivity

Definition: Protocol $A$ is noninteractive if all users speak once, simultaneously and independently.

Make all randomizer assignments beforehand.
Types of LDP Interactivity

Definition: Protocol $\mathcal{A}$ is sequentially interactive [DJW13] if all users speak once (possibly in multiple rounds).

Make randomizer assignments adaptively.
Types of LDP Interactivity

Definition: Protocol $A$ is \textit{fully interactive} if users may interact arbitrarily (possibly speak multiple times, in multiple rounds).
Types of LDP Interactivity

Noninteractive

Sequentially Interactive

Fully Interactive

Local Differential Privacy
Types of LDP Interactivity

Noninteractive

Sequentially Interactive

Fully Interactive

\begin{align*}
\text{Noninteractive} & : & x_1 & \rightarrow y_1 \\
& & x_2 & \rightarrow y_2 \\
& & x_3 & \rightarrow y_3 \\
\text{Sequentially Interactive} & : & x_1 & \rightarrow y_1 & \rightarrow y_2 & \rightarrow y_3 \\
& & y_1 & \rightarrow y_2 & \rightarrow y_3 \\
\text{Fully Interactive} & : & x_1 & \rightarrow y_{1,1} & \rightarrow y_{1,3} & \rightarrow y_3 \\
& & x_2 & \rightarrow y_{2,2} & \rightarrow y_3 \\
& & x_3 & \rightarrow y_{3,4} & \rightarrow y_3
\end{align*}

\# rounds = 1

\# rounds \leq \# users

\# rounds = ???

Local Differential Privacy
Types of LDP Interactivity

Noninteractive

Sequentially Interactive

Fully Interactive

Local Differential Privacy

[KLNRS08] [DF19]
Types of LDP Interactivity

Noninteractive

Sequentially Interactive

Fully Interactive

This Work

Local Differential Privacy

[x_1, x_2, x_3] → [y_1, y_2, y_3]

[DF18] [KLNRS08]
Outline

1. Preliminaries

2. **Tool**: LDP $\approx$ Noisy Communication

3. Application: Exponential Separation
Tool: LDP \approx\text{ Noisy Communication}

General connection between two-party communication complexity (CC) and multi-party locally private sample complexity (SC).
Tool: LDP $\approx$ Noisy Communication

General connection between two-party communication complexity ($CC$) and multi-party locally private sample complexity ($SC$).

Two-party problem: Alice has input $a$, Bob has input $b$, want to compute some function of $a$ and $b$. 
Tool: LDP $\approx$ Noisy Communication

General connection between two-party communication complexity ($CC$) and multi-party locally private sample complexity ($SC$).

Two-party problem: Alice has input $a$, Bob has input $b$, want to compute some function of $a$ and $b$.

Multi-party problem: each user randomly gets $a$ or $b$, want to compute some function of $a$ and $b$. 
Tool: LDP $\approx$ Noisy Communication

**Theorem:** Given two-party problem $P_2$ and multi-party analogue $P_m$, for $\varepsilon = O(1)$, $\text{SC}^{\varepsilon,S}(P_m) = \Theta(\text{CC}(P_2)/\varepsilon^2)$. 
Theorem: Given two-party problem $P_2$ and multi-party analogue $P_m$, for $\varepsilon = O(1)$, $SC^{\varepsilon,S}(P_m) = \Theta(CC(P_2)/\varepsilon^2)$. 

Tool: LDP $\approx$ Noisy Communication
Theorem: Given two-party problem $P_2$ and multi-party analogue $P_m$, for $\varepsilon = O(1)$, $SC^{\varepsilon,S}(P_m) = \Theta(CC(P_2)/\varepsilon^2)$. 

Tool: LDP \approx Noisy Communication
Tool: LDP $\approx$ Noisy Communication

Proof Sketch
Three parts:
Tool: LDP ≈ Noisy Communication

Proof Sketch
Three parts:

CC($P_2$) $\leftrightarrow$ $\varepsilon'$-noisy CC($P_2$) $\leftrightarrow$ 1-bit SC$_{\varepsilon,S}(P_m)$ $\leftrightarrow$ SC$_{\varepsilon,S}(P_m)$
Proof Sketch
Three parts:

1. Connect \textbf{CC} of noiseless and \textbf{CC} of $\varepsilon'$-noisy two-party communication problems [BR14, BM15]
Tool: LDP \approx \text{Noisy Communication}

Proof Sketch

Three parts:

1. Connect $\text{CC}$ of noiseless and $\text{CC}$ of $\varepsilon'$-noisy two-party communication problems [BR14, BM15]

2. Connect $\text{CC}$ of $\varepsilon'$-noisy two-party and $\text{SC}$ of 1-bit-per-person sequentially interactive $\varepsilon$-locally private multi-party communication problems
Tool: LDP $\approx$ Noisy Communication

Proof Sketch

2.: $\rightarrow$

$\star \mathbf{P}_{m}^{\varepsilon,S,1} \rightarrow \mathbf{P}_{2}^{\varepsilon',\text{noisy}}$: Alice and Bob randomly partition users in $\mathbf{P}_{m}^{\varepsilon,S,1}$ between them
Proof Sketch

2. $\rightarrow$

* $P_{m}^{\varepsilon,S,1} \rightarrow P^{\varepsilon',\text{noisy}}$: Alice and Bob randomly partition users in $P_{m}^{\varepsilon,S,1}$ between them.

* for each bit in $P_{m}^{\varepsilon,S,1}$, Alice or Bob sends bit with probabilities calibrated to $\varepsilon'$ and current randomizer.

Tool: LDP \(\approx\) Noisy Communication
Proof Sketch

2.: →

* $P_{\varepsilon,S,1}^m \rightarrow P_{\varepsilon',\text{noisy}}^2$ : Alice and Bob randomly partition users in $P_{\varepsilon,S,1}^m$ between them

* for each bit in $P_{\varepsilon,S,1}^m$ Alice or Bob sends bit with probabilities calibrated to $\varepsilon'$ and current randomizer

* # bits for $P_{\varepsilon',\text{noisy}}^2 = #$ users for $P_{\varepsilon,S,1}^m$
Proof Sketch

2.: \( P_{m}^{\epsilon, S, 1} \rightarrow P_{2}^{\epsilon', \text{noisy}} \): Alice and Bob randomly partition users in \( P_{m}^{\epsilon, S, 1} \) between them.

* for each bit in \( P_{m}^{\epsilon, S, 1} \) Alice or Bob sends bit with probabilities calibrated to \( \epsilon' \) and current randomizer.

* # bits for \( P_{2}^{\epsilon', \text{noisy}} \) = # users for \( P_{m}^{\epsilon, S, 1} \)
Proof Sketch

2.: →

* $P_{m}^{\varepsilon,S,1} \rightarrow P_{2}^{\varepsilon',noisy}$: Alice and Bob randomly partition users in $P_{m}^{\varepsilon,S,1}$ between them.

* for each bit in $P_{m}^{\varepsilon,S,1}$, Alice or Bob sends bit with probabilities calibrated to $\varepsilon'$ and current randomizer.

* # bits for $P_{2}^{\varepsilon',noisy} = # users for P_{m}^{\varepsilon,S,1}$

uses $SC^{\varepsilon,S} = # randomizer calls$

may fail for $SC^{\varepsilon,F}$
Proof Sketch

2. \( \leftarrow * \quad P^m \quad \epsilon, S, 1 \leftarrow P_2^{\epsilon',\text{noisy}} \): for each bit in \( P_2^{\epsilon',\text{noisy}} \), draw a new user
Proof Sketch
2.: ←
* $\begin{array}{l}
\text{\textbf{P}}_{m}^{\varepsilon, S, 1} \leftarrow \text{\textbf{P}}_{2}^{\varepsilon', \text{noisy}}
\end{array}$: for each bit in $\text{\textbf{P}}_{2}^{\varepsilon', \text{noisy}}$, draw a new user
* user sends bit through $\varepsilon$-RR if correct of Alice and Bob otherwise uniform random
The tool is LDP $\approx$ Noisy Communication

Proof Sketch
Three parts:

1. Connect $\text{CC}$ of noiseless and $\text{CC}$ of $\epsilon'$-noisy two-party communication problems [BR14, BM15]

2. Connect $\text{CC}$ of $\epsilon'$-noisy two-party and $\text{SC}$ of 1-bit-per-person sequentially interactive $\epsilon$-locally private multi-party communication problems
Tool: LDP $\approx$ Noisy Communication

Proof Sketch
Three parts:

1. Connect $\text{CC}$ of noiseless and $\text{CC}$ of $\varepsilon'$-noisy two-party communication problems [BR14, BM15]

2. Connect $\text{CC}$ of $\varepsilon'$-noisy two-party and $\text{SC}$ of 1-bit-per-person sequentially interactive $\varepsilon$-locally private multi-party communication problems

3. Connect $\text{SC}$ of 1-bit-per-person and $\text{SC}$ of generic sequentially interactive $\varepsilon$-locally private multi-party communication problems [BS15]
Tool: LDP $\approx$ Noisy Communication

Proof Sketch
Three parts:

- $\text{CC}(P_2)$
- $\epsilon'$-noisy $\text{CC}(P_2)$
- 1-bit $\text{SC}^{\epsilon,S}(P_m)$
- $\text{SC}^{\epsilon,S}(P_m)$

Tool: LDP $\approx$ Noisy Communication
Tool: LDP $\approx$ Noisy Communication

Theorem: Given two-party problem $P_2$ and multi-party analogue $P_m$, for $\varepsilon = O(1)$, $SC^{\varepsilon,S}(P_m) = \Theta(CC(P_2)/\varepsilon^2)$. 
Theorem: Given two-party problem $P_2$ and multi-party analogue $P_m$, for $\varepsilon = O(1)$, $\text{SC}^{\varepsilon,S}(P_m) = \Theta(\text{CC}(P_2)/\varepsilon^2)$.

Can now get multi-party $\text{SC}^{\varepsilon,S}$ lower bounds straight from two-party $\text{CC}$ lower bounds.
Outline

1. Prelims

2. Tool: LDP $\approx$ Noisy Communication

3. Application: Exponential Separation
Application: Exponential Separation

Useful because two-party $\text{CC}$ lower bounds are well-studied.

Lemma [GKR16]: Solving the two-party hidden layers problem requires $\text{CC} = \Omega(2^k)$.
Application: Exponential Separation

Goal: find path through tree that is consistent with $a$ and $b$ layers
Application: Exponential Separation

Goal: find path through tree that is consistent with \( a \) and \( b \) layers

---

Application: Exponential Separation
Application: Exponential Separation

Goal: find path through tree that is consistent with $a$ and $b$ layers
Goal: find path through tree that is consistent with \(a\) and \(b\) layers
Application: Exponential Separation

Goal: find path through tree that is consistent with $a$ and $b$ layers
Application: Exponential Separation

Goal: find path through tree that is consistent with \( a \) and \( b \) layers
Application: Exponential Separation

Goal: find path through tree that is consistent with \(a\) and \(b\) layers

\[2^k\] \(-\text{ary}

\[2^{2^k}\] \text{layers}

Application: Exponential Separation
Application: Exponential Separation

Useful because two-party $CC$ lower bounds are well-studied.

**Lemma [GKR16]:** Solving the two-party *hidden layers* problem requires $CC = \Omega(2^k)$.

**Corollary:** For $P^m$ = multi-party hidden layers problem, $SC^{\varepsilon,S}(P^m) = \Omega(2^k/\varepsilon^2)$. 
Application: Exponential Separation

Useful because two-party \( \text{CC} \) lower bounds are well-studied.

**Lemma [GKR16]:** Solving the two-party hidden layers problem requires \( \text{CC} = \Omega(2^k) \).

**Corollary:** For \( P_m \) = multi-party hidden layers problem, \( SC^{\varepsilon,S}(P_m) = \Omega(2^k/\varepsilon^2) \). But \( SC^{\varepsilon,F}(P_m) = O(k/\varepsilon^2) \).
Application: Exponential Separation

Useful because two-party $\text{CC}$ lower bounds are well-studied.

**Lemma [GKR16]:** Solving the two-party *hidden layers* problem requires $\text{CC} = \Omega(2^k)$.

**Corollary:** For $P_m^m$ = multi-party hidden layers problem, $SC^{\epsilon,S}(P_m^m) = \Omega(2^k/\epsilon^2)$. But $SC^{\epsilon,F}(P_m^m) = O(k/\epsilon^2)$.

(At each node, for all $2^k$ possible next nodes, ask all users if correct. Can handle $2^k$ by union bound on RR accuracy.)
“The extra effort is worth it if we’re trying to solve the hidden layers problem!

Surgeon General
Jerome Adams

Dep. Surgeon General
Erica Schwartz
“The extra effort is worth it if we’re trying to solve the hidden layers problem!

And we can prove it using a general connection between local differential privacy and communication complexity.”
"The extra effort is worth it if we’re trying to solve the hidden layers problem! And we can prove it using a general connection between local differential privacy and communication complexity."

"Great! We’ll keep that in mind if we ever need to solve the hidden layers problem."

Surgeon General Jerome Adams

Dep. Surgeon General Erica Schwartz
Open Questions

- How large can the gap between SI and FI be?
Open Questions

- How large can the gap between SI and FI be?
- Separation for “natural” problem?
Open Questions

- How large can the gap between SI and FI be?
- Separation for “natural” problem?
- How powerful are FI protocols?
Open Questions

● How large can the gap between SI and FI be?
● Separation for “natural” problem?
● How powerful are FI protocols?

arxiv.org/abs/1907.00813
References

5. [DMNS06] “Calibrating Noise to Sensitivity in Private Data Analysis”. Dwork, Mcsherry, Nissim, Smith. TCC.
8. [KLNRS08] “What Can We Learn Privately?”. Kasiviswanathan, Lee, Nissim, Raskhodnikova, Smith. STOC.