Computational Bottlenecks in Differential Privacy

Jonathan Ullman, Northeastern University

Bar Ilan University Winter School: Differential Privacy. February 13, 2017.

Outline

- Computational hardness results in DP
  - Surprising tradeoffs between privacy, utility, and computational efficiency
  - Interesting cryptographic techniques: digital signatures, traitor-tracing schemes, watermarking

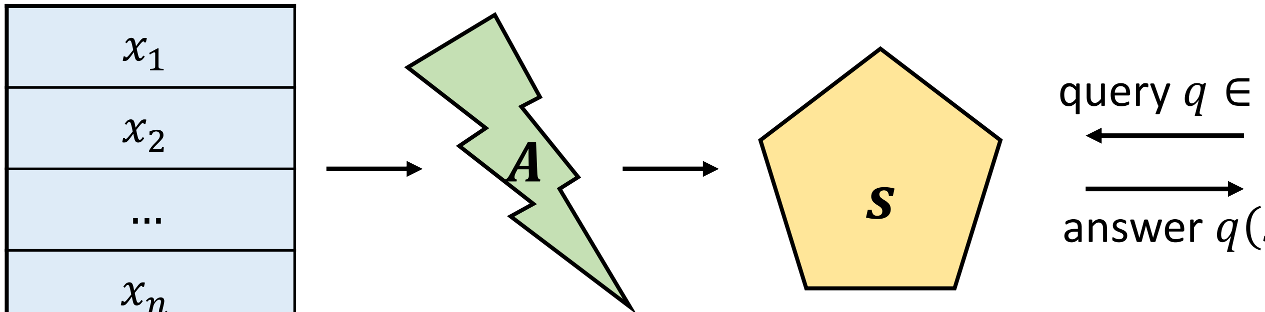
Query Release Review

dataset
$x \in \{0,1\}^{n\times d}$

(ε,δ)-dp
algorithm

"summary" of x

family of poly-time
statistical queries
$Q = \{q_1, q_2, \dots\}$



query $q \in Q$

answer $q(s)$

$$q(x) = \frac{1}{n}\sum_i \phi(x_i)$$

s is accurate if
$$\max_{q\in Q} |q(x) - q(s)| \le .01$$

Laplace Mechanism:

- Adds error $\tilde{O}\left(\frac{\sqrt{|Q|}}{\varepsilon n}\right)$; limited to $\approx n^2$ queries
- Running time is $\mathrm{poly}(n, d, |q_1| + |q_2| + \cdots)$
- Summary is just a list of noisy answers

PMW Mechanism:

- Adds error $O\left(\frac{\sqrt{d}\cdot \ln|Q|}{\varepsilon n}\right)^{1/2}$; can answer $\approx 2^{n/\sqrt{d}}$ queries
- Running time is $\mathrm{poly}\left(n, 2^d, |q_1| + |q_2| + \cdots\right)$
- Summary is a *synthetic dataset* $\hat{x} \in \{0,1\}^{n\times d}$

Main Questions

1. Can we answer $\gg n^2$ statistical queries privately, accurately, and in $\text{poly}(n, d)$ time?

2. Can we efficiently generate private synthetic datasets?

Laplace Mechanism:

- Adds error $\tilde{O}\left(\frac{\sqrt{|Q|}}{\varepsilon n}\right)$; limited to $\approx n^2$ queries
- Running time is $\text{poly}(n, d, |q_1| + |q_2| + \cdots)$
- Summary is just a list of noisy answers

PMW Mechanism:

- Adds error $O\left(\frac{\sqrt{d} \cdot \ln |Q|}{\varepsilon n}\right)^{1/2}$; can answer $\approx 2^{n/\sqrt{d}}$ queries
- Running time is $\text{poly}\left(n, 2^d, |q_1| + |q_2| + \cdots\right)$
- Summary is a *synthetic dataset* $\hat{x} \in \{0,1\}^{n \times d}$

Hardness of Large Query Families

Assuming OWF

Theorem*:
There is a family of 2^d statistical queries Q on $\{0,1\}^d$ s.t. no DP algorithm can take a dataset of size $n = \mathrm{poly}(d)$, run in time $\mathrm{poly}(n, d)$, and output an accurate summary for Q.

Compare to Private Multiplicative Weights, which can answer any 2^d queries over the universe $\{0,1\}^d$ in time $\mathrm{poly}(n, 2^d)$ given a dataset of size $O(d^{3/2})$.

*[Dwork+'09, Boneh-Zhandry'14, Kowalczyk+'17]

Traitor-Tracing Schemes

users $1, \ldots, n$
secret keys $sk_i \in \{0,1\}^{\ell(key)}$

can encrypt a message $b \in \{0,1\}$
so that every user can decrypt

broadcaster

sk_1

sk_i

sk_n

$c = Enc(mk, b) \in \{0,1\}^{\ell(ctext)}$

$\Downarrow$

$\forall i \in [n] \quad Dec(sk_i, c) = b$

master key $mk \in \{0,1\}^*$

$P\big(Enc(b)\big) = b$

$Trace_{mk}$

$\{sk_i\}_{i \in C} \rightarrow$ $\rightarrow i \in U$

coalition of users
$U \subseteq \{1, \ldots, n\}$

efficient pirate
decoder

tracing
algorithm

Correctness =
Security =

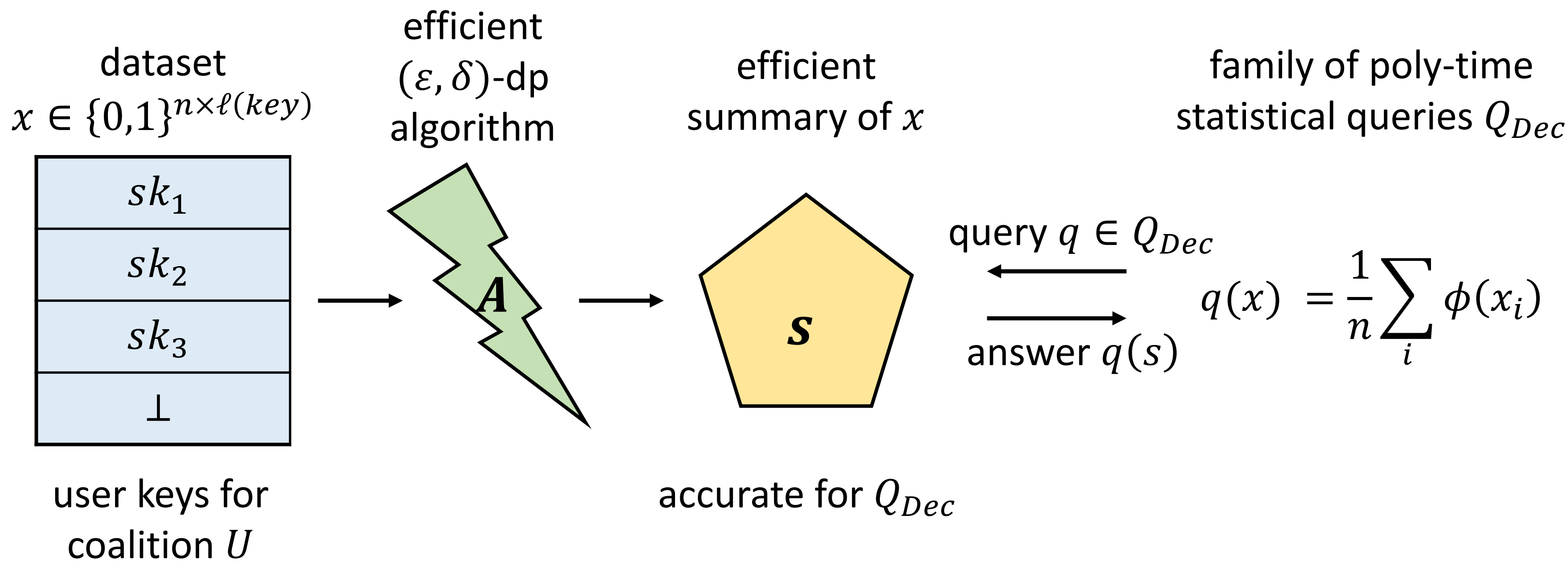
Traitor Tracing vs. Differential Privacy

Theorem*:
If there is a TTS for n users then there is a family of $2^{\ell(ctext)}$ statistical queries Q over $\{0,1\}^{\ell(key)}$ such that no DP algorithm can take a dataset of size n, run in polynomial time, and output an accurate summary for Q.

Number of users	$\Leftrightarrow$	Dataset size
Number of ciphertexts	$\Leftrightarrow$	Number of queries
Length of secret keys	$\Leftrightarrow$	Length of dataset elements
Efficient pirate decoder	$\Leftrightarrow$	Efficient, accurate summary

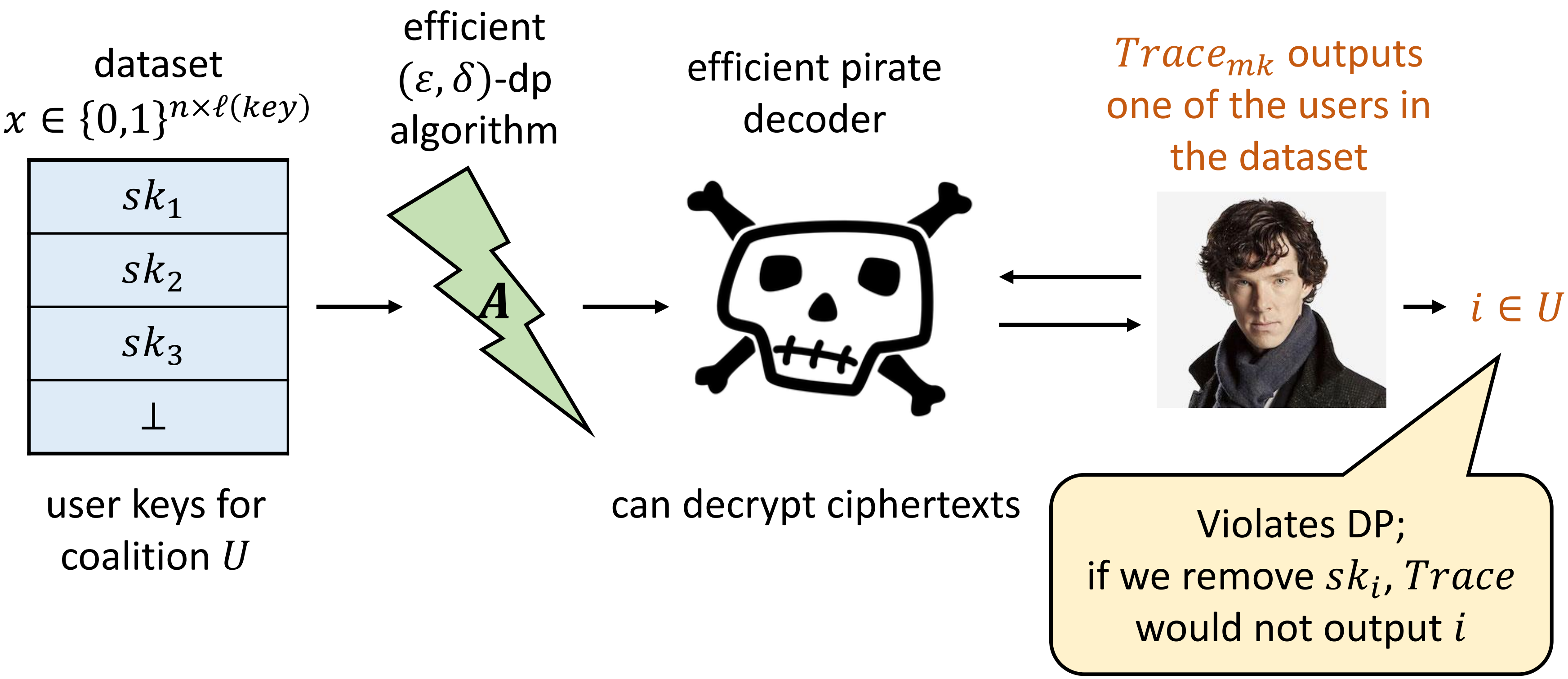
*[Dwork+'09]

Traitor Tracing vs. Differential Privacy



- Defining the queries:
  - $Q_{Dec} = \left\{ q_c \mid c \in \{0,1\}^{\ell(ctext)} \right\}$, where $q_c(sk) = Dec(sk, c)$
  - If $c = Enc(mk, b)$ then $q_c(x) = \frac{1}{n} \sum_i q_c(sk_i) = \frac{1}{n} \sum_i Dec(sk_i, c) = b$
  - So an accurate summary for Q_{Dec} can be used to decrypt ciphertexts!

Traitor Tracing vs. Differential Privacy



- Defining the queries:
  - $Q_{Dec} = \{ q_c \mid c \in \{0,1\}^{\ell(ctext)} \}$, where $q_c(sk) = Dec(sk, c)$
  - If $c = Enc(mk, b)$ then $q_c(x) = \frac{1}{n} \sum_i q_c(sk_i) = \frac{1}{n} \sum_i Dec(sk_i, c) = b$
  - So an accurate summary for Q_{Dec} can be used to decrypt ciphertexts!

Traitor Tracing vs. Differential Privacy

Theorem*:
If there is a TTS for n users then there is a family of $2^{\ell(ctext)}$ statistical queries Q over $\{0,1\}^{\ell(key)}$ such that no DP algorithm can take a dataset of size n, run in polynomial time, and output an accurate summary for Q.

Number of users	$\Leftrightarrow$	Dataset size
Number of ciphertexts	$\Leftrightarrow$	Number of queries
Length of secret keys	$\Leftrightarrow$	Length of dataset elements
Efficient pirate decoder	$\Leftrightarrow$	Efficient, accurate summary

*[Dwork+'09]

Traitor Tracing vs. Differential Privacy

Theorem*:
If there is a TTS for n users then there is a family of $2^{\ell(ctext)}$ statistical queries Q over $\{0,1\}^{\ell(key)}$ such that no DP algorithm can take a dataset of size n, run in polynomial time, and output an accurate summary for Q.

Theorem [BZ'14, KMU'17]:
Assuming OWF, for every d, and every $n = \text{poly}(d)$, there is a "good enough" TTS with $\ell(key) = \ell(ctext) = d$ secure against $\text{poly}(d)$ time adversaries.

Hardness of Large Query Families

Assuming OWF

Theorem*:
There is a family of 2^d statistical queries Q on $\{0,1\}^d$ s.t. no DP algorithm can take a dataset of size $n = \text{poly}(d)$, run in time $\text{poly}(n, d)$, and output an accurate summary for Q.

Compare to Private Multiplicative Weights, which can answer any 2^d queries over the universe $\{0,1\}^d$ in time $\text{poly}(n, 2^d)$ given a dataset of size $O(d^{3/2})$.

*[Dwork+'09, Boneh-Zhandry'14, Kowalczyk+'17]

Hardness of Large Query Families

Assuming OWF

Theorem*:

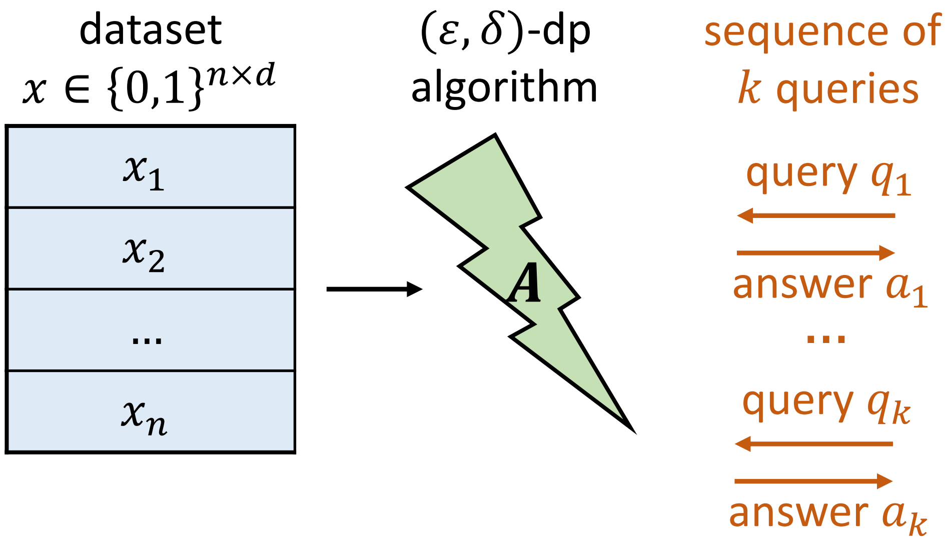
There is a family of 2^d statistical queries Q on $\{0,1\}^d$ s.t. no DP algorithm can take a dataset of size $n = \text{poly}(d)$, run in time $\text{poly}(n, d)$, and output an accurate summary for Q.

Way stronger assumption

Theorem [KMUZ'16]:

- Exists a hard family of $O(n^7)$ queries over $\{0,1\}^d$
  - Small family of queries, large data universe
- Exists a hard family of 2^d queries over $\{1, \ldots, O(n^7)\}$
  - Large family of queries, small data universe

Interactive Mechanisms



answers accurate if
$$\max_j \left|q_j(x) - a_j\right| \leq .01$$

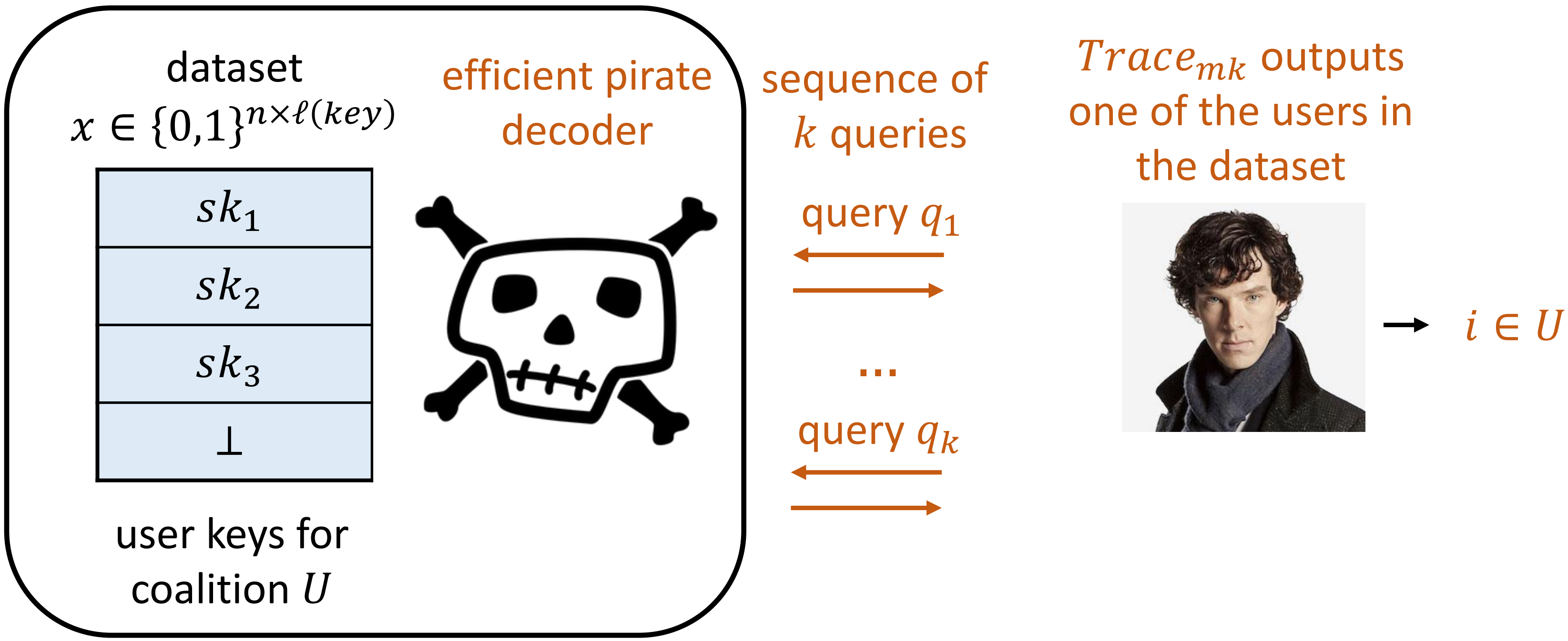
Laplace Mechanism:

- Adds error $\tilde{O}\left(\frac{\sqrt{k}}{\varepsilon n}\right)$; limited to n^2 queries
- Running time is $\text{poly}(n, d, |q|)$ per query

PMW Mechanism:

- Adds error $O\left(\frac{\sqrt{d} \cdot \ln(k)}{\varepsilon n}\right)^{1/2}$; can answer $2^{n/\sqrt{d}}$ queries
- Running time is $\text{poly}\left(n, 2^d, |q|\right)$ per query

Interactive Mechanisms



- Changes in the interactive setting:
  - Same family of queries
  - View $A(x)$ together as the efficient pirate decoder
  - Relevant measure is now the number of queries made by $Trace_{mk}$

Interactive Mechanisms

Theorem [U'13]:
If there is a TTS for n users with keys in $\{0,1\}^{\ell(key)}$ such that $Trace$ makes k queries, then no efficient DP interactive mechanism answers k arbitrary queries.

Theorem [U'13]:
Assuming OWF, for every ℓ, and every $n = \text{poly}(\ell)$, there is a "good enough" TTS that makes $k = \tilde{O}(n^2)$ queries and is secure against $\text{poly}(\ell)$ time adversaries

*"Good enough" means that the scheme traces "stateful-but-cooperative" pirates.

Hardness of Large Query Families

Assuming OWF

<u>Theorem [U'13]:</u>
No DP algorithm can take a dataset $x \in \{0,1\}^{n \times d}$, run in time $poly(n, d, |q|)$ per query, and accurately answer $k = \tilde{O}(n^2)$ arbitrary statistical queries

Compare to Laplace, which answers $k = \tilde{\Omega}(n^2)$ queries in time $\mathrm{poly}(n, d, |q|)$ per query.

Compare to Private Multiplicative Weights, which answers $k \approx 2^{n/\sqrt{d}}$ queries in time $poly\left(n, 2^d, |q|\right)$ per query.

Hardness of Large Query Families

Assuming OWF

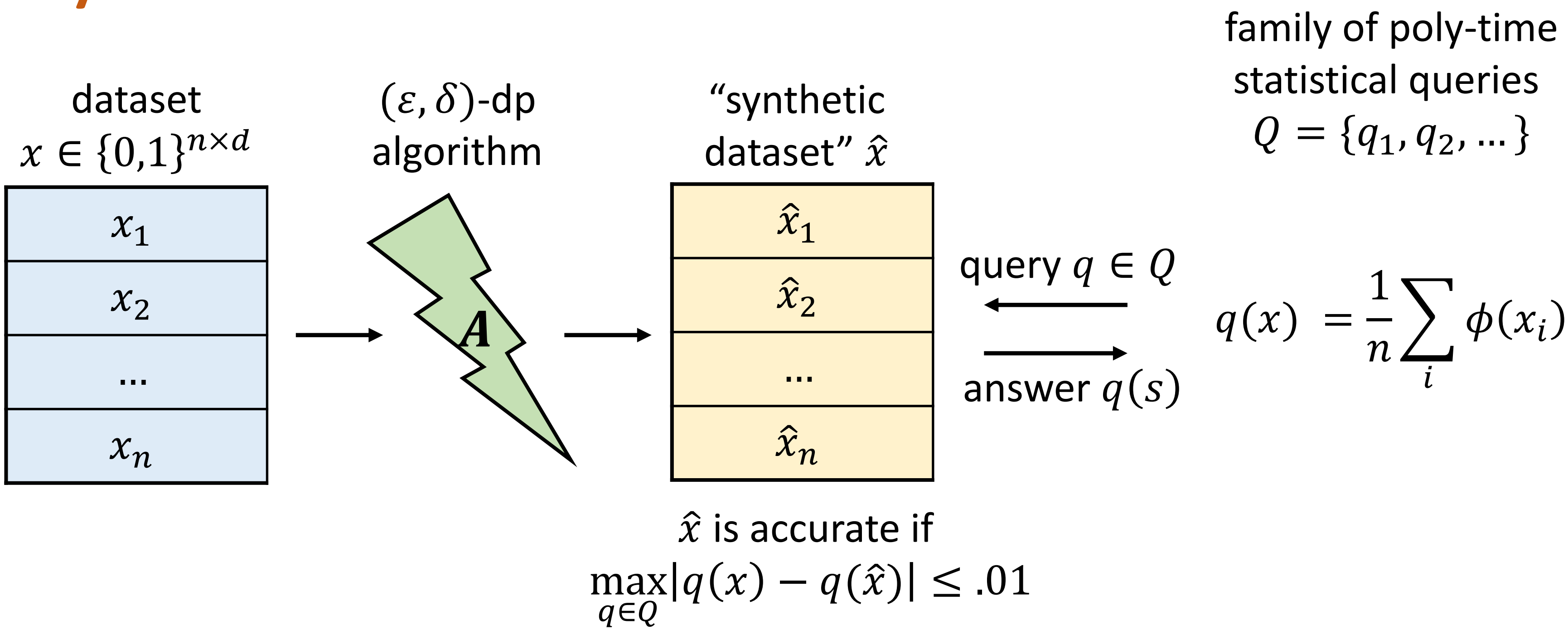
Theorem [U'13]:

No DP algorithm can take a dataset $x \in \{0,1\}^{n \times d}$, run in time $poly(n, d, |q|)$ per query, and accurately answer $k = \tilde{O}(n^2)$ arbitrary statistical queries

Previous results apply to arbitrary---and, statisticians might say, rather funny looking---statistical queries.

What can we say about *simple* families of queries?

Synthetic Datasets



PMW Mechanism:

- Adds error $O\left(\frac{\sqrt{d} \cdot \ln |Q|}{\varepsilon n}\right)^{1/2}$; can answer $2^{n/\sqrt{d}}$ queries
- Running time is $\mathrm{poly}\left(n, 2^d, |q_1| + |q_2| + \cdots\right)$
- Summary is a *synthetic dataset* $\hat{x} \in \{0,1\}^{n \times d}$

Hardness of Synthetic Data

Assuming OWF

<u>Theorem [DNRRV'09, UV'11]:</u>
No DP algorithm can take a dataset of size $n = \mathrm{poly}(d)$, run in time $\mathrm{poly}(n, d)$, and output a synthetic dataset accurate for the means of and correlations between each column.

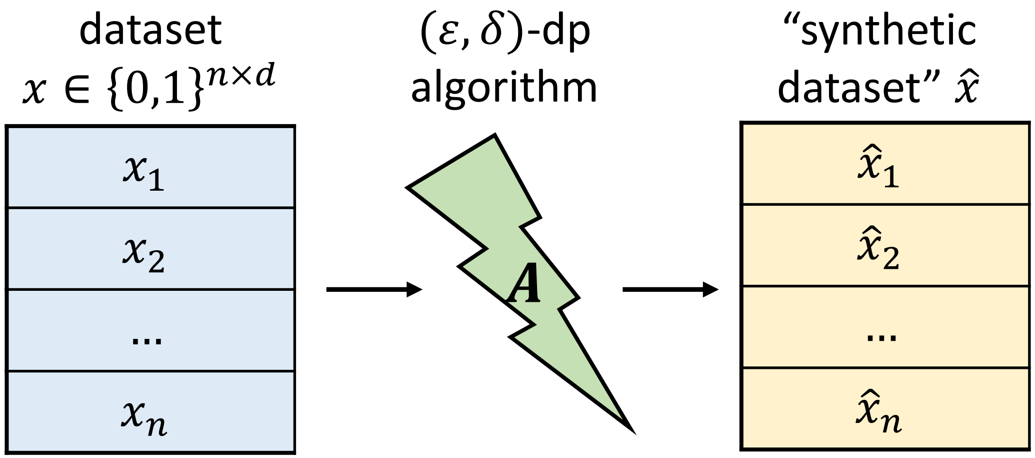
d^2 statistical queries of the form

$$q_{i,k}(x) = \frac{1}{n} \sum_{i} x_{ij} \cdot x_{ik}$$

Laplace is efficient and accurate, but no synthetic data.

PMW is accurate and generates synthetic data, but requires at least 2^d time.

Hardness of Synthetic Data



family of queries
$Q = \{q_1, q_2, \dots\}$

$\hat{x}$ is accurate if
$$\max_{q \in Q} |q(x) - q(\hat{x})| \leq .01$$

Digital Signatures:

- Three algorithms $(Gen, Sign, Ver)$
- $Gen \to (sk, vk) \in \{0,1\}^d$
- For a message $m \in \{0,1\}^d$, $Sign(sk, m) \to \sigma \in \{0,1\}^d$
- $Ver\big(vk, (m, \sigma)\big) \in \{0,1\}$; outputs 1 if $\sigma = Sign(sk, m)$
- No poly(d) time adversary, even one with a signing oracle, can forge a new pair (m^*, σ^*) s.t. $Ver\big(vk, (m^*, \sigma^*)\big) = 1$

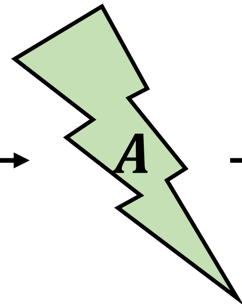
Hardness of Synthetic Data

dataset
$x \in \{0,1\}^{n \times 2d}$

(m_1, σ_1)
(m_2, σ_2)
…
(m_n, σ_n)

random
signed messages

(ε, δ)-dp
algorithm



"synthetic
dataset" $\hat{x}$

$\hat{x}_1$
$\hat{x}_2$
…
$\hat{x}_n$

define the query
$q_{vk}(m, \sigma) = Ver(vk, (m, \sigma))$

$\exists\, (m^*, \sigma^*) \in \hat{x}$
$Ver(vk, (m^*, \sigma^*)) = 1$

$q(x) = 1$, so if $\hat{x}$ is accurate
$q_{vk}(\hat{x}) \geq .99$

Argument:

- Choose $(sk, vk) \leftarrow Gen$
- Let x be n random message-signature pairs
- Query: "what fraction of this dataset is valid signatures?"
- Accuracy implies that the dataset contains a valid signature
  - Case 1: $(m^*, \sigma^*) \in x$: violates privacy
  - Case 2: $(m^*, \sigma^*) \notin x$: violates unforgeability

Hardness of Synthetic Data

Assumes that secure cryptography is possible.

Theorem [DNRRV'09]:
No DP algorithm can take a dataset of size $n = \mathrm{poly}(d)$, run in time $\mathrm{poly}(n, d)$, and output a synthetic dataset accurate for all "verification queries" $Q_{vk} = \{Ver(vk,\cdot)\}_{vk\in\{0,1\}^d}$

- Can reduce the *number* of queries to by embedding the verification key in the dataset.
- Can *simplify* the queries to means and correlations using techniques from hardness of approximation
  - Encodings of the signed messages as probabilistically checkable proofs (PCPs)

Hardness of Synthetic Data

Assuming OWF

<u>Theorem [DNRRV'09, UV'11]:</u>
No DP algorithm can take a dataset of size $n = \mathrm{poly}(d)$, run in time $\mathrm{poly}(n, d)$, and output a synthetic dataset accurate for the means of and correlations between each column.

d^2 statistical queries of the form

$$q_{i,k}(x) = \frac{1}{n} \sum_i x_{ij} \cdot x_{ik}$$

Outline

- Computational hardness results in DP
  - Surprising tradeoffs between privacy, utility, and computational efficiency
  - Interesting cryptographic techniques: digital signatures, traitor-tracing schemes, watermarking
- Hardness of private data release
- Hardness of generating synthetic data