Make Up Your Mind: The Price of Online Queries in Differential Privacy

or: An Excuse to Survey Differential Privacy Lower Bounds

Taiwan Theory Day

May 17, 2016

Mark Bun Harvard U.

Thomas Steinke Harvard U.

Jonathan Ullman Northeastern U.

The Challenge of Data Privacy

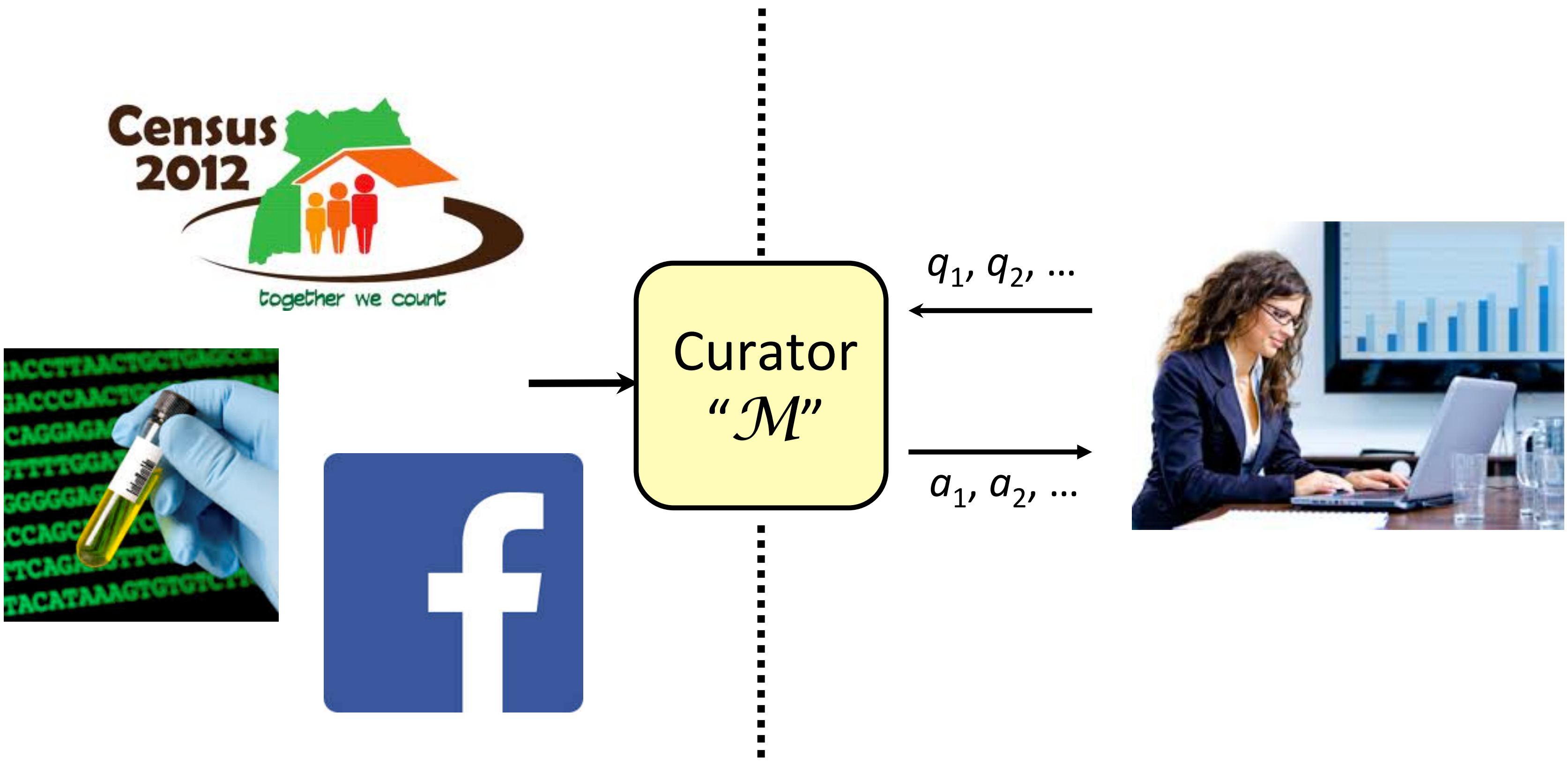
Protect privacy of subjects

Enable useful statistical analyses

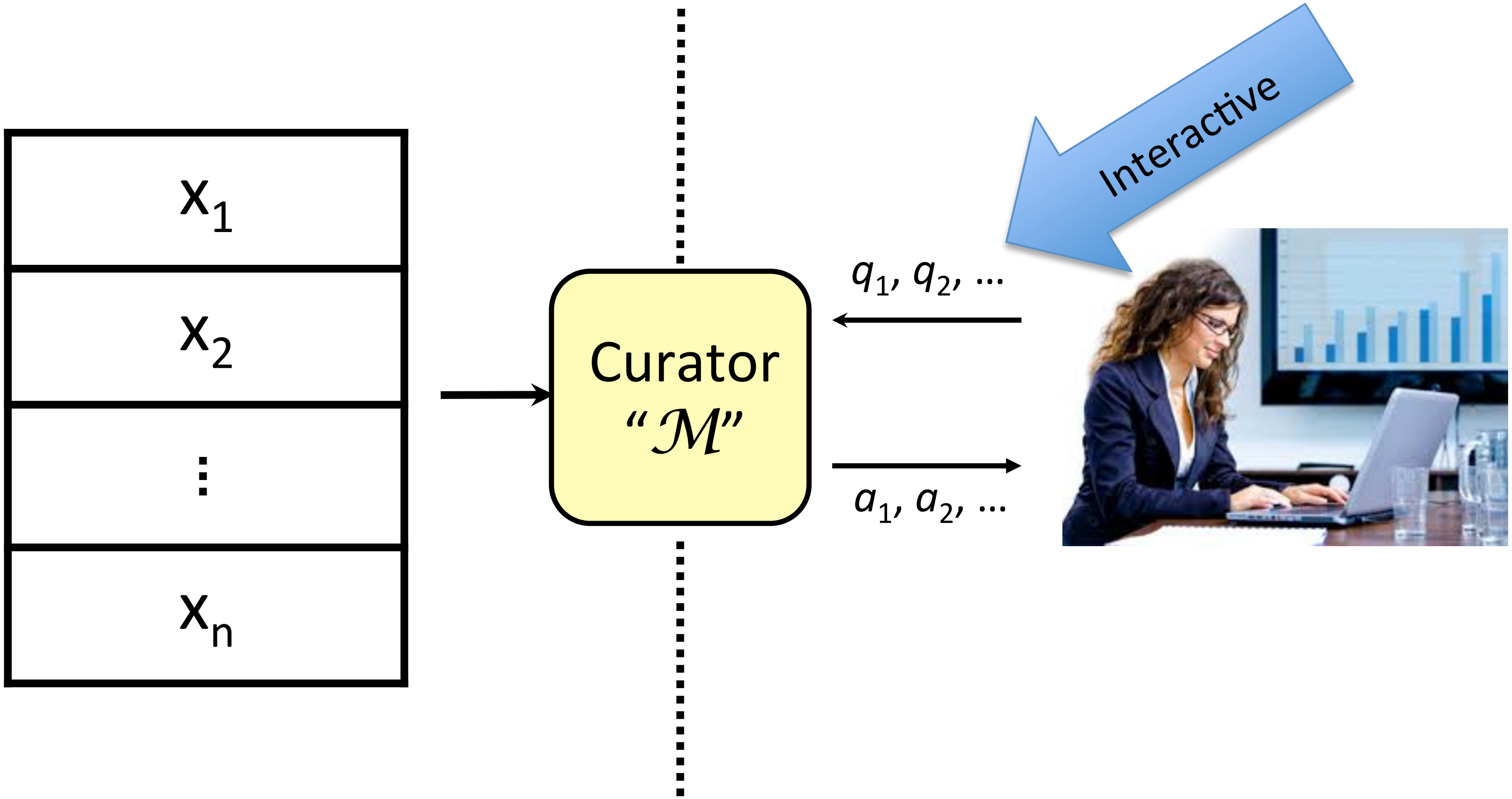
Census Data
Medical Records
Social Networks
Location Data

Social science research
Identifying genes
Replicating studies
Recommendation services

Privacy-Preserving Data Analysis



Privacy-Preserving Data Analysis



How Should We Model Interaction?

- “Offline”: Analyst chooses all of her queries in advance and receives answers together
- “Adaptive”: Analyst chooses/asks queries one at a time

  …or another possibility?

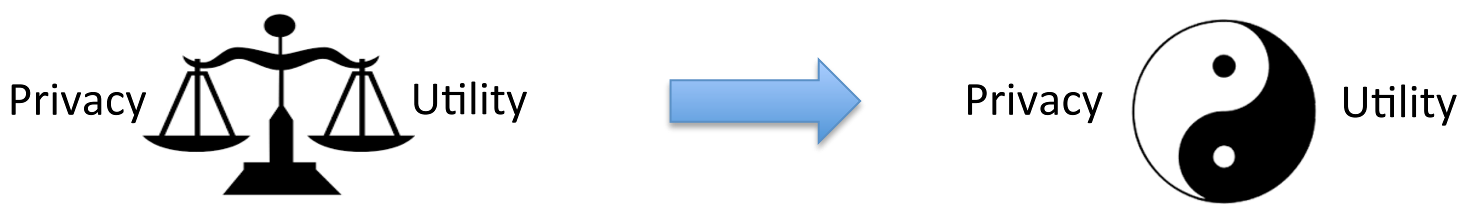
- **This work:** How does changing the model of interaction affect what can be accomplished with differential privacy?

Why Might This Matter?

- Rich theory of differential privacy – sophisticated algorithms matched by strong lower bounds

- Differential privacy prevents false discovery, even in *adaptive* data analysis

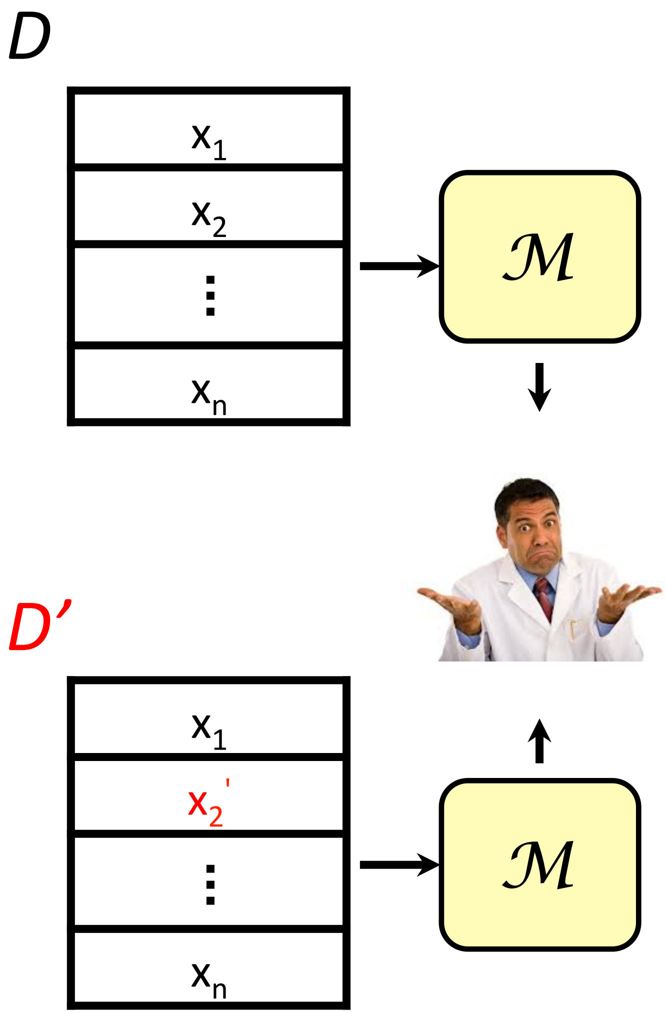
  [Dwork-Feldman-Hardt-Pitassi-Reingold-Roth14, Hardt-Ullman14]



Does handling adaptivity in DP really come for free?

Differential Privacy

Dinur-Nissim03+Dwork, Dwork-Nissim04, Blum-Dwork-McSherry-Nissim05,
Dwork-McSherry-Nissim-Smith06, Dwork-Kenthapadi-McSherry-Mironov-Naor06



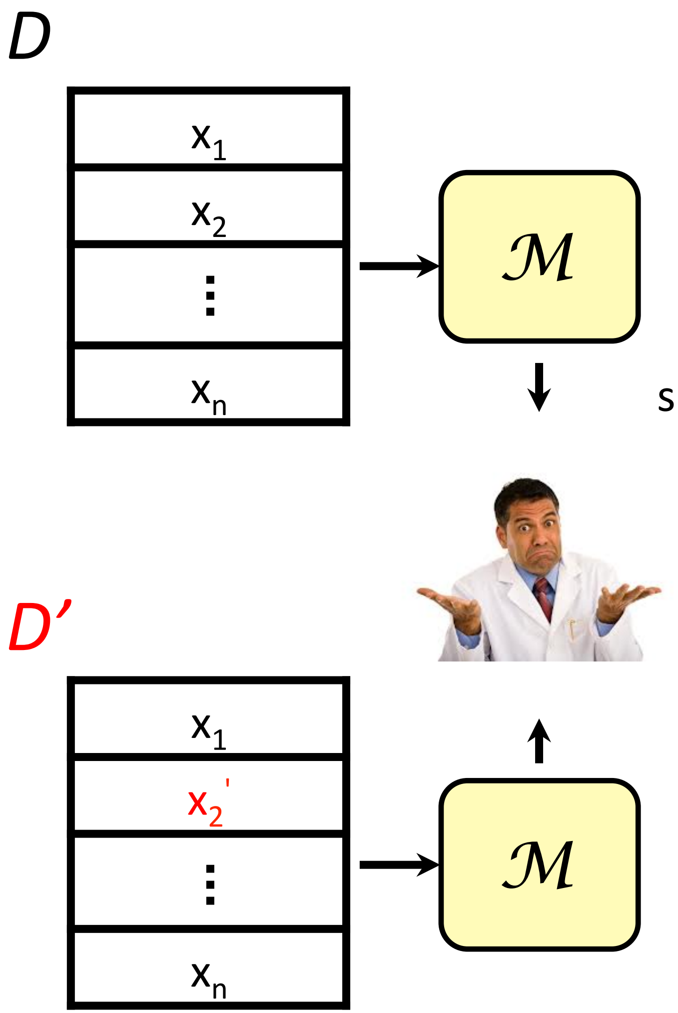
D and *D′* are **neighbors** if they differ on one row

$\mathcal{M}$ is **differentially private** if for all neighbors *D*, *D′*:

$$\mathcal{M}(D) \approx \mathcal{M}(D')$$

Differential Privacy

Dinur-Nissim03+Dwork, Dwork-Nissim04, Blum-Dwork-McSherry-Nissim05,
Dwork-McSherry-Nissim-Smith06, Dwork-Kenthapadi-McSherry-Mironov-Naor06



D and *D′* are **neighbors** if they differ on one row

small const., e.g. ε = 0.1

"cryptographically small"
require δ << 1/*n*, often δ = negl(*n*)

$\mathcal{M}$ is **(ε,δ)-differentially private** if for all neighbors *D*, *D′* and S⊆Range($\mathcal{M}$):

$$\Pr[\mathcal{M}(D')\in S] \le (1+\varepsilon)\Pr[\mathcal{M}(D)\in S] + \delta$$

Counting Queries

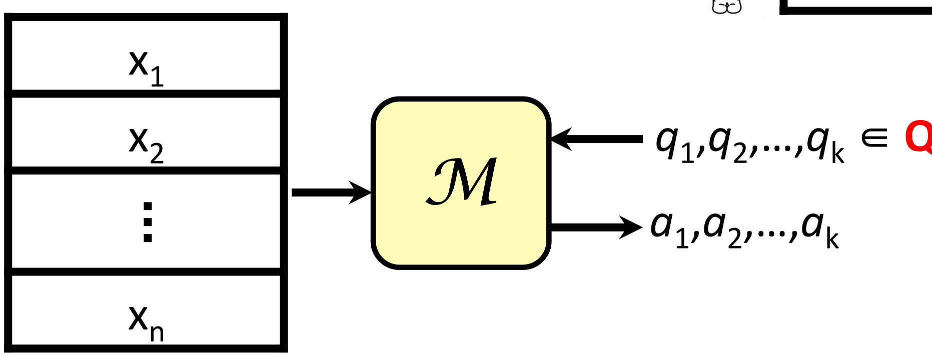
"What fraction of the rows of *D* satisfy some property *q*?"

E.g. **attribute means**

q = HasMouth?

$q(D) = 2/4$

HasMouth?	Bakes?	Clothed?	OnJet?
0	1	1	1
1	0	1	1
1	0	0	1
0	1	1	1



$\mathcal{M}$ is **accurate for *k* queries from Q** if

$|a_i - q_i(D)| < 0.05$ for every i

(with high probability)

(Privately) Answering Attribute Means

[DN03, DN04, BDMN05, DMNS06]

d binary attributes

n people

HasMouth?	Bakes?	Clothed?	OnJet?
0	1	1	1
1	0	1	1
1	0	0	1
0	1	1	1

1/2
+
Noise(   )

(Privately) Answering Attribute Means

[DN03, DN04, BDMN05, DMNS06]

d binary attributes

n people

	HasMouth?	Bakes?	Clothed?	OnJet?
	0	1	1	1
	1	0	1	1
	1	0	0	1
	0	1	1	1

1/2
+
Noise(O(1/n))

(Privately) Answering Attribute Means

[DN03, DN04, BDMN05, DMNS06]

d binary attributes

n people

	HasMouth?	Bakes?	Clothed?	OnJet?
	0	1	1	1
	1	0	1	1
	1	0	0	1
	0	1	1	1
	1/2 + Noise( )	1/2 + Noise( )	3/4 + Noise( )	1 + Noise( )

Disclaimer: This talk hides all polylogs

(Privately) Answering Attribute Means

[DN03, DN04, BDMN05, DMNS06]

d binary attributes

n people

HasMouth?	Bakes?	Clothed?	OnJet?
0	1	1	1
1	0	1	1
1	0	0	1
0	1	1	1
1/2 + Noise($O(d^{1/2}/n)$)	1/2 + Noise($O(d^{1/2}/n)$)	3/4 + Noise($O(d^{1/2}/n)$)	1 + Noise($O(d^{1/2}/n)$)

Non-trivial accuracy requires $d < n^2$

$\Rightarrow$ can answer $k = d = \Omega(n^2)$ queries

Disclaimer: This talk hides all polylogs

Not Just Attribute Means

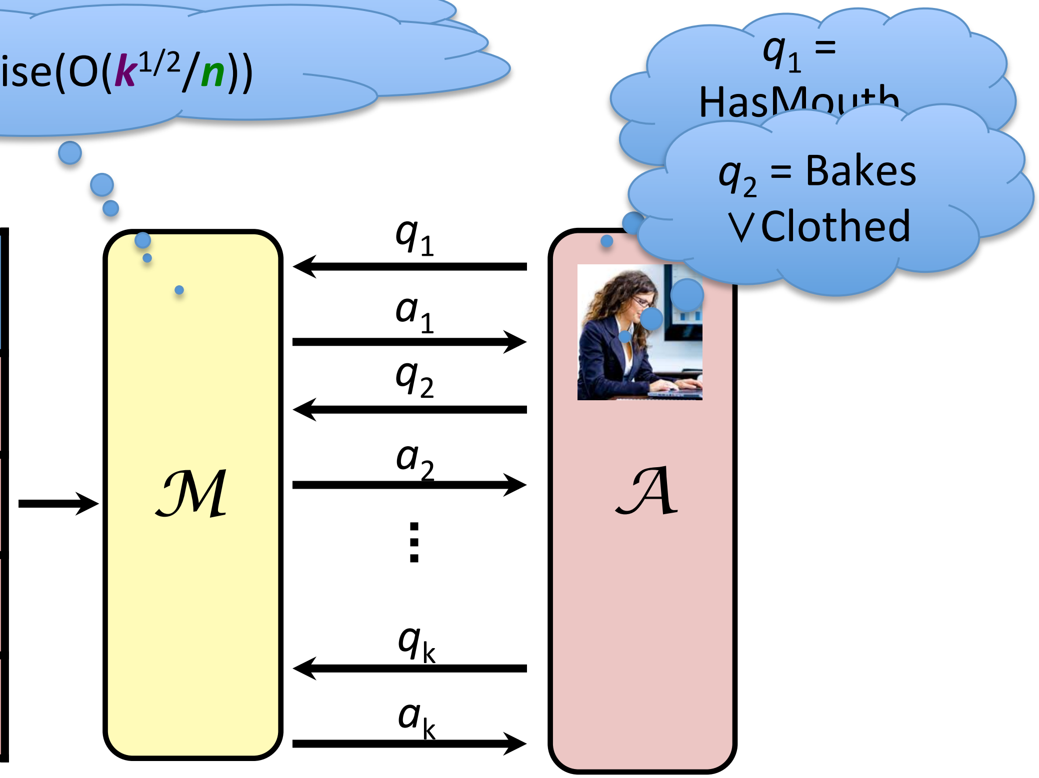
[DN03, DN04, BDMN05, DMNS06]

$a_2 = q_2(D) + \text{Noise}(O(\boldsymbol{k}^{1/2}/\boldsymbol{n}))$

d binary attributes

n ppl

Has Mouth	Bakes	Clothed	OnJet
0	1	1	1
1	0	1	1
1	0	0	1
0	1	1	1



Can answer $\boldsymbol{k} = \Omega(\boldsymbol{n}^2)$ *adaptively chosen* counting queries

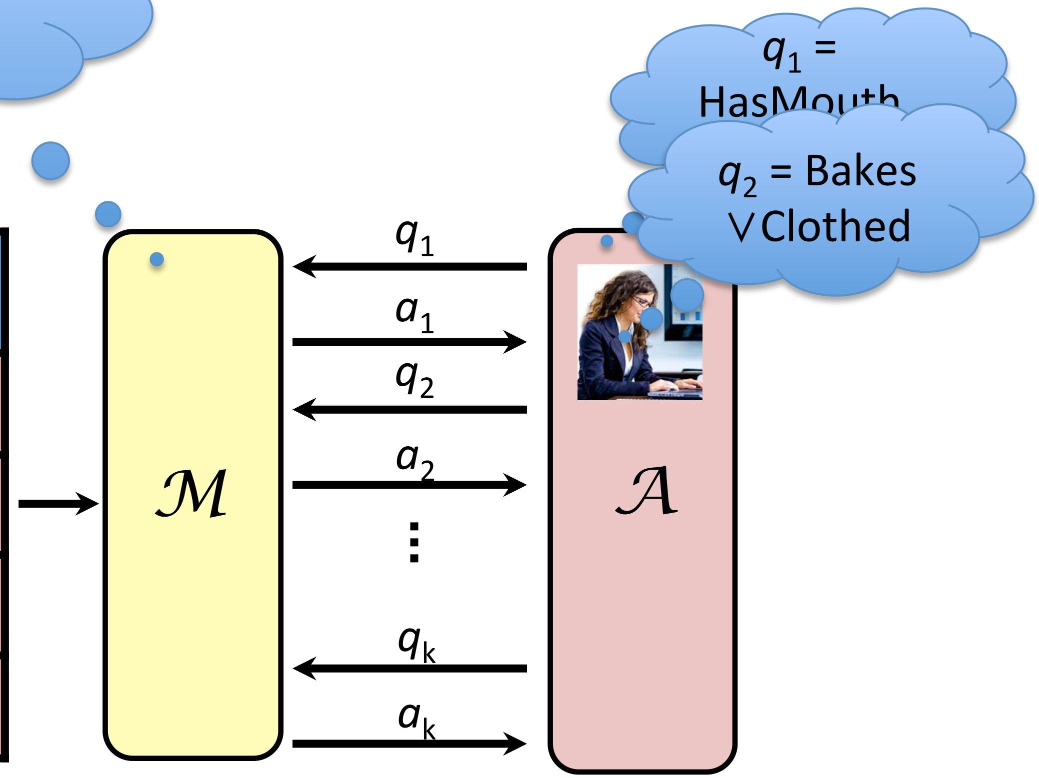
…And Not Just n^2 Queries

[Blum-Ligett-Roth08, Roth-Roughgarden10, Hardt-Rothblum10]

d binary attributes

n ppl

Has Mouth	Bakes	Clothed	OnJet
0	1	1	1
1	0	1	1
1	0	0	1
0	1	1	1



"Private Multiplicative Weights" [Hardt-Rothblum10]

Can answer $\boldsymbol{k} = \exp(\Omega(\boldsymbol{n}/\boldsymbol{d}^{1/2}))$ *adaptively chosen* counting queries

(= exponentially many queries when $\boldsymbol{n} >> \boldsymbol{d}^{1/2}$)

(Counting)

How Many Queries Can We Answer?

($\varepsilon = 0.1$, $\delta = o(1/n)$) – differential privacy

Upper bound: $n \ll d^{1/2}$
(Independent Noise)

Upper bound: $n \gg d^{1/2}$
("Advanced Algorithms")

Adaptive

Adaptive
$\forall \mathbf{Q}$: $k = \Omega(n^2)$ [...DMNS06]
$\forall \mathbf{Q}$: $\exp(\Omega(n/d^{1/2}))$ [HR10]

Matching Lower Bounds

- Can't answer more than $\boldsymbol{k}$ = exp(O($\boldsymbol{n}$)) queries

  [Dinur-Nissim03]

"Reconstruction Attack"

log $\boldsymbol{n}$ bits | 1 bit

$\boldsymbol{n}$ ppl

Public "ID"		Sensitive "b"
0	0	1
0	1	0
1	0	0
1	1	1

Ask *all* 2^{n} counting queries of the form:
$q_S(x) = (x_{ID} \in S) \wedge x_b$ where $S \subseteq \{0,1\}^{\log n}$

Reconstruct any database D' with
$|q_S(D') - q_S(D)| < 0.05$ for all q_S

<u>Claim:</u> (0.05)-accurate answers $\Rightarrow$
b' agrees with b in 80% of entries

<u>Proof:</u> If $|b' - b|_1 > 0.2$, then
$|q_S(D') - q_S(D)| > 0.1$ for either
$S^{>} = \{i : b_i > b_i'\}$ or $S^{<} = \{i : b_i < b_i'\}$

Matching Lower Bounds

- Can't answer more than $\boldsymbol{k}$ = exp(O($\boldsymbol{n}$)) queries
  [Dinur-Nissim03]

  …

- Independent noise is tight for attribute means:
  Can only answer O($\boldsymbol{n}^2$) queries [B.-Ullman-Vadhan14]
- Private mult. weights is tight for conjunctions:
  Can only answer exp(O($\boldsymbol{n}/\boldsymbol{d}^{1/2}$)) queries [B.-Ullman-Vadhan14]

- All lower bounds apply to a *fixed set* of queries

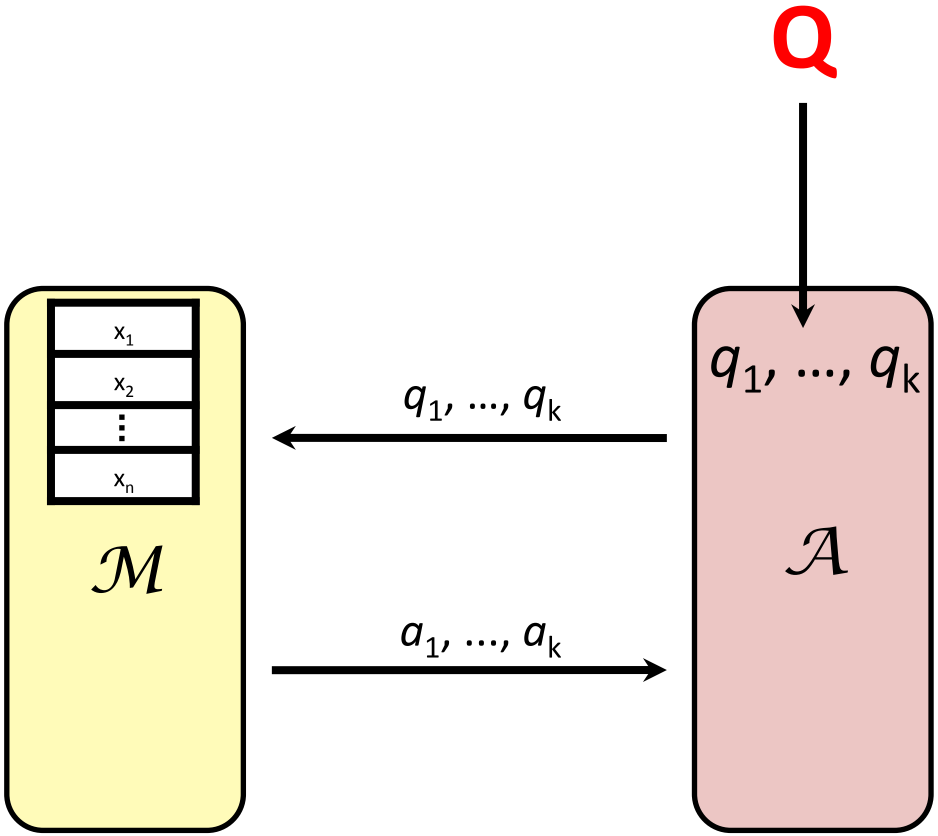
(Counting)

How Many Queries Can We Answer?

($\varepsilon = 0.1$, $\delta = o(1/n)$) – differential privacy	Offline	Adaptive
Upper bound: $n \ll d^{1/2}$ (Independent Noise)	←	$\forall Q$: $k = \Omega(n^2)$ [...DMNS06]
Upper bound: $n \gg d^{1/2}$ ("Advanced Algorithms")	←	$\forall Q$: $\exp(\Omega(n/d^{1/2}))$ [HR10]
Lower bound: $n \ll d^{1/2}$ (Attribute Means)	$\exists Q$: $O(n^2)$ [BUV14]	→
Lower bound: $n \gg d^{1/2}$ (Conjunctions)	$\exists Q$: $\exp(O(n/d^{1/2}))$ [BUV14]	→

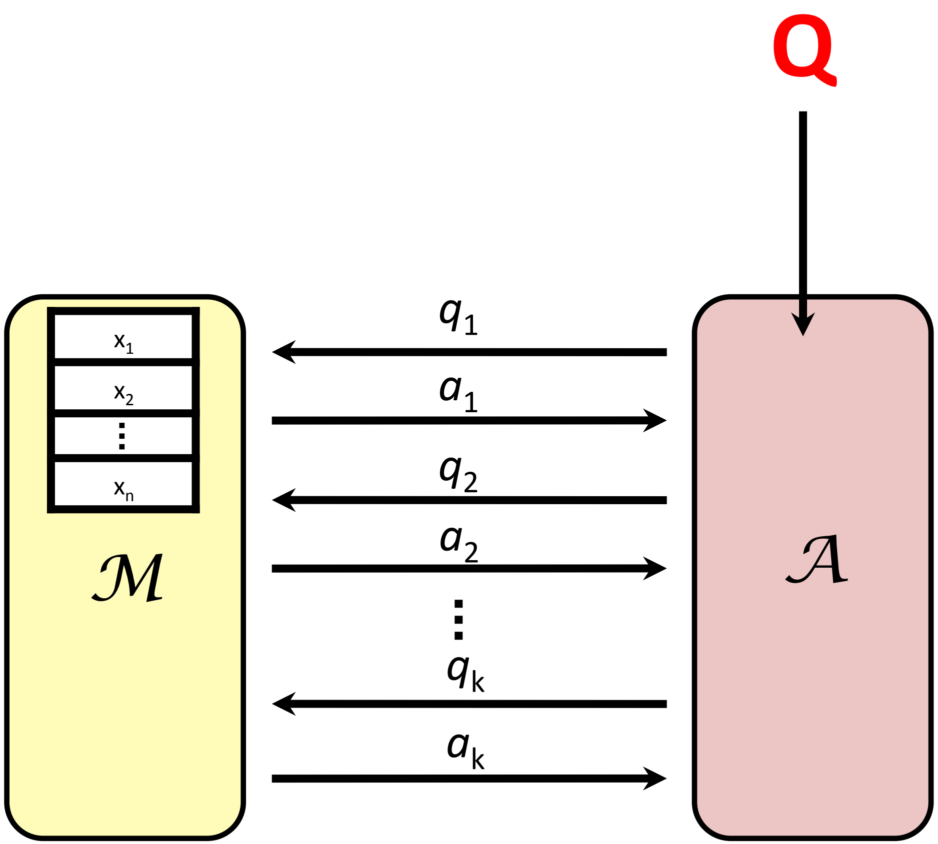
Question: Are these models equivalent?

The **OFFline** Model



1. $\mathcal{A}$ chooses ***k*** queries $q_1, \ldots, q_k$ from **Q**
2. $\mathcal{A}$ gives queries to $\mathcal{M}$ in a single batch
3. $\mathcal{M}$ releases answers $a_1,\ldots,a_k$

The **Adaptive** Model



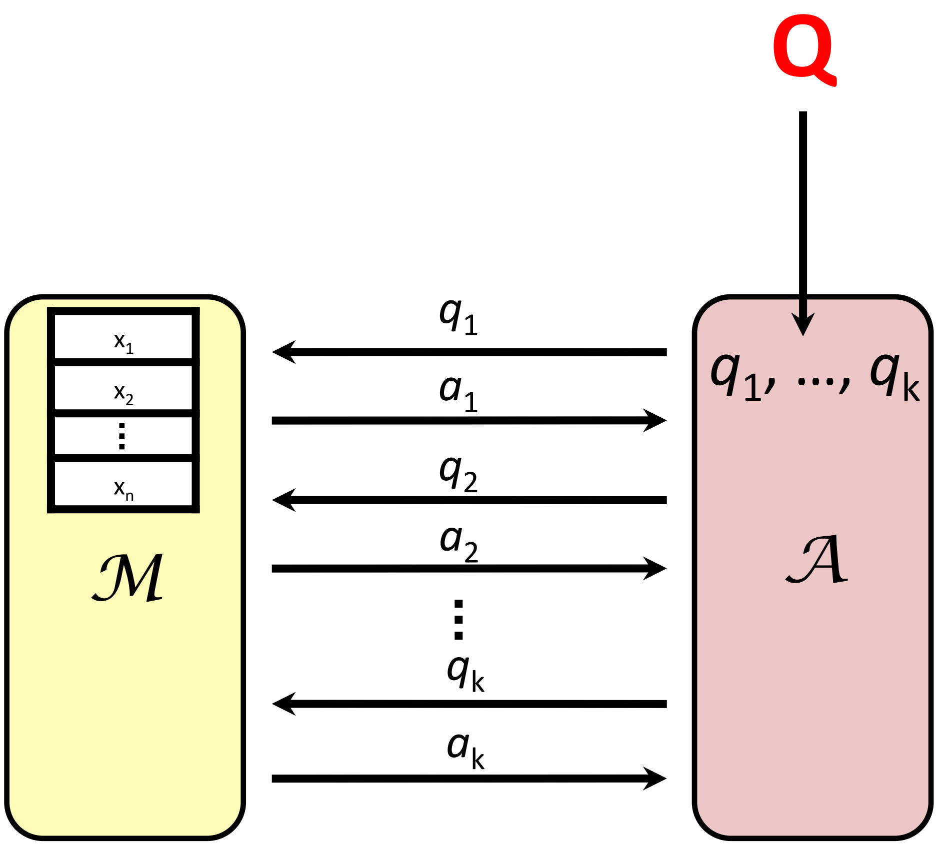
In each round j = 1,…,***k***:

1. $\mathcal{A}$ chooses a query q_j (depending on $q_1, a_1, \ldots, q_{j-1}, a_{j-1}$)

2. $\mathcal{M}$ must release a_j before seeing q_{j+1}

The **ONline** Model

(Non-adaptive)



1. $\mathcal{A}$ chooses ***k*** queries $q_1, \ldots, q_k$ from **Q**

2. In each round j = 1,…,***k***:

   $\mathcal{M}$ must release a_j before seeing q_{j+1}

Our Results

All three models are distinct

- **Offline ≠ Online**

Family $\mathbf{Q}_{\mathbf{prefix}}$ of counting queries

Offline: Can answer $\boldsymbol{k} = \exp(\Omega(\boldsymbol{n}^{1/2}))$ queries

Online: Can only answer $\boldsymbol{k} = O(\boldsymbol{n}^2)$ queries

- **Online ≠ Adaptive**

Family $\mathbf{Q}_{\mathbf{corr}}$ of “search” queries

Online: $\boldsymbol{k} = \exp(\Omega(\boldsymbol{n}))$ queries

Adaptive: $\boldsymbol{k} = O(1)$ queries

Offline vs. Online

"Prefix queries"

$\mathbf{Q_{prefix}} = \{\ q_S : \{0,1\}^{\boldsymbol{d}} \ \rightarrow\ \{0,1\}\ \}$

For $S = \{y_1,\ldots,y_m \in \{0,1\}^{\leq \boldsymbol{d}} : m \leq \boldsymbol{d}\}$ and $x \in \{0,1\}^{\leq \boldsymbol{d}}$:

Define $q_S(x) = 1$ iff $\exists y \in S$ that is a prefix of x

Example

$S = \{0, 10, 001, 110\} \subseteq \{0,1\}^{\leq 4}$

$x = 1010 \in \{0,1\}^{\leq 4}$

Offline vs. Online

“Prefix queries”

$\mathbf{Q}_{\text{prefix}} = \{\ q_S : \{0,1\}^{d} \rightarrow \{0,1\}\ \}$

For $S = \{y_1,\ldots,y_m \in \{0,1\}^{\leq d} : m \leq d\}$ and $x \in \{0,1\}^{\leq d}$:

Define $q_S(x) = 1$ iff $\exists y \in S$ that is a prefix of x

Example

$S = \{0, \mathbf{10}, 001, 110\} \subseteq \{0,1\}^{\leq 4}$

$x = \mathbf{10}10 \in \{0,1\}^{\leq 4}$

$\Rightarrow \quad q_S(x) = 1$

Offline vs. Online

"Prefix queries"

$\mathbf{Q_{prefix}} = \{\ q_S : \{0,1\}^{\boldsymbol{d}} \ \rightarrow\ \{0,1\}\ \}$

For $S = \{y_1,\dots,y_m \in \{0,1\}^{\leq \boldsymbol{d}} : m \leq \boldsymbol{d}\}$ and $x \in \{0,1\}^{\leq \boldsymbol{d}}$:

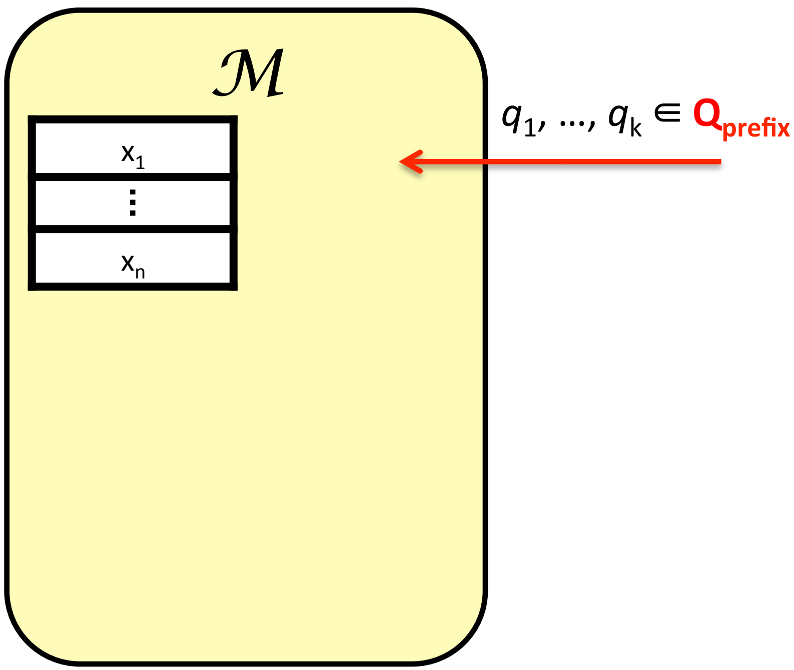
Define $q_S(x) = 1$ iff $\exists y \in S$ that is a prefix of x

Intuition for separation

Offline: Structure of queries enables dimensionality reduction

Online: As hard as *attribute means*

An Offline Algorithm

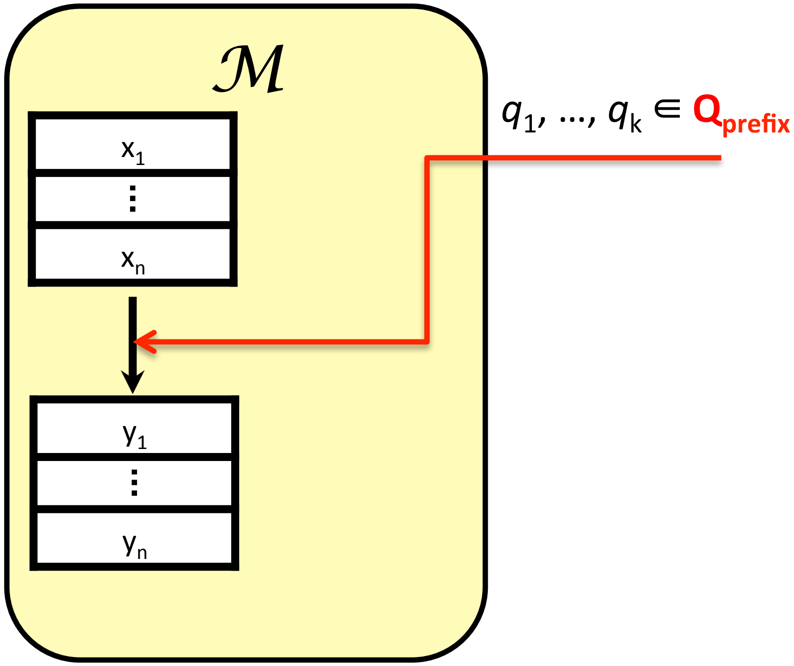


<u>Algorithm $\mathcal{M}$</u>

Input: queries $q_1,\ldots,q_k$ corresponding to sets $S_1,\ldots,S_k$

1. Let $S = S_1 \cup S_2 \cup \ldots \cup S_k$
2. Replace each x_i with longest $y_i \in S$ which is a prefix of x_i
3. Run your favorite "advanced algorithm" on $(y_1,\ldots,y_n)$

An Offline Algorithm

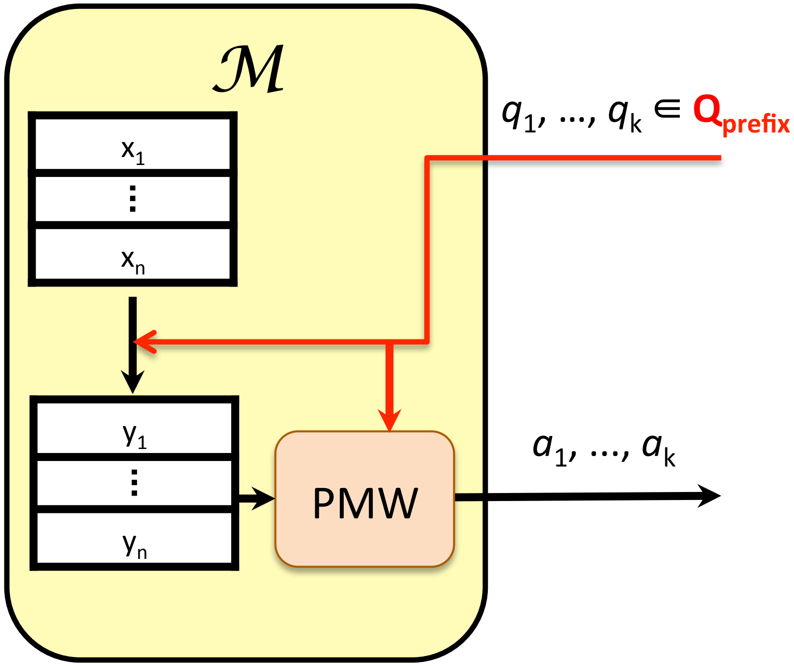


<u>Algorithm $\mathcal{M}$</u>

Input: queries $q_1,\ldots,q_k$ corresponding to sets $S_1,\ldots,S_k$

1. Let $S = S_1 \cup S_2 \cup \ldots \cup S_k$
2. Replace each x_i with longest $y_i \in S$ which is a prefix of x_i
3. Run your favorite "advanced algorithm" on $(y_1,\ldots,y_n)$

An Offline Algorithm

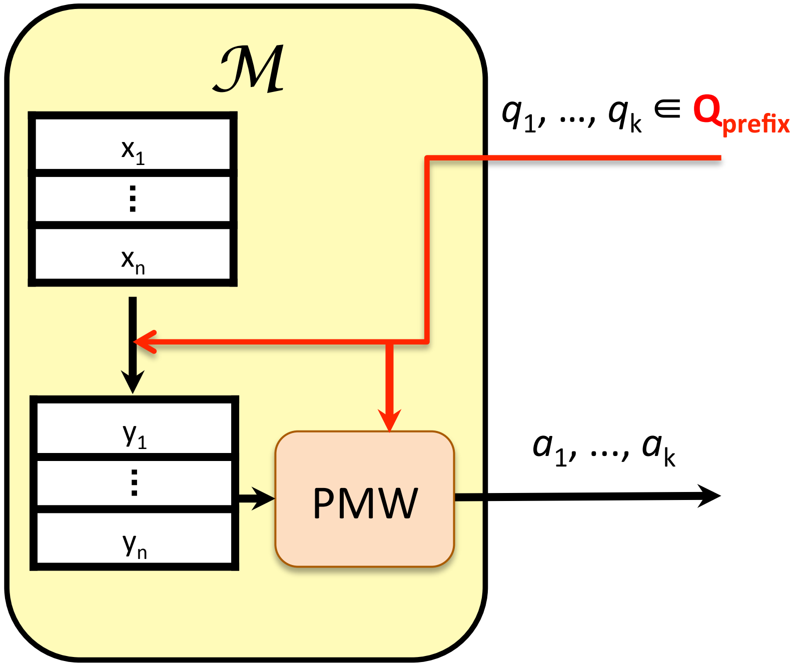


<u>Algorithm $\mathcal{M}$</u>

Input: queries $q_1,\ldots,q_k$ corresponding to sets $S_1,\ldots,S_k$

1. Let $S = S_1 \cup S_2 \cup \ldots \cup S_k$
2. Replace each x_i with longest $y_i \in S$ which is a prefix of x_i
3. Run your favorite "advanced algorithm" on $(y_1,\ldots,y_n)$

An Offline Algorithm



<u>Algorithm $\mathcal{M}$</u>

Input: queries $q_1,\ldots,q_k$ corresponding to sets $S_1,\ldots,S_k$

1. Let $S = S_1 \cup S_2 \cup \ldots \cup S_k$
2. Replace each x_i with longest $y_i \in S$ which is a prefix of x_i
3. Run your favorite "advanced algorithm" on $(y_1,\ldots,y_n)$

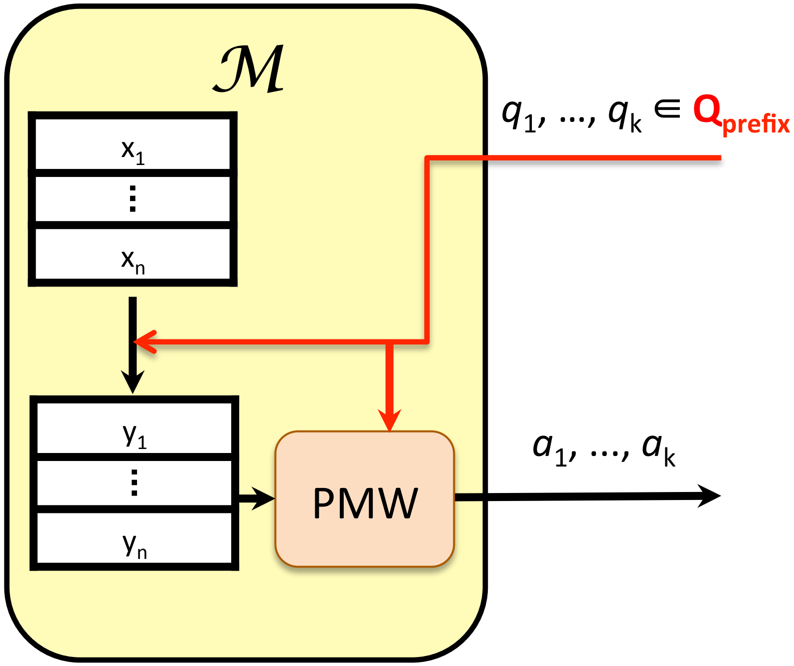
<u>Example:</u>

$x_1 = 1110$
$x_2 = 0010$
$x_3 = 0101$
$x_4 = 0110$

$\in (\{0,1\}^{\leq 4})^4$

$S_1 = \{1\}$
$S_2 = \{01, 10\}$
$S_3 = \{001, 011\}$

An Offline Algorithm



<u>Algorithm $\mathcal{M}$</u>

Input: queries $q_1,\ldots,q_k$ corresponding to sets $S_1,\ldots,S_k$

1. Let $S = S_1 \cup S_2 \cup \ldots \cup S_k$
2. Replace each x_i with longest $y_i \in S$ which is a prefix of x_i
3. Run your favorite "advanced algorithm" on $(y_1,\ldots,y_n)$

<u>Example:</u>

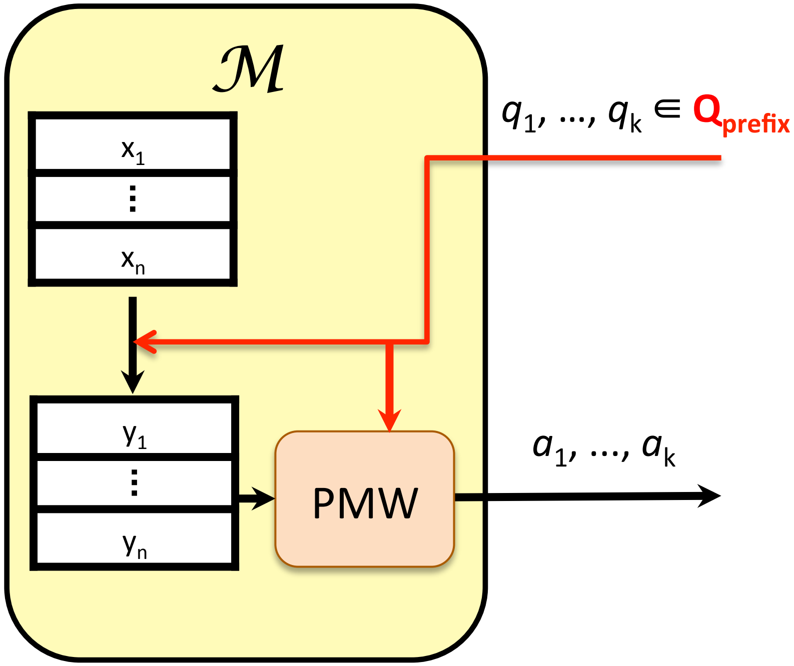
$x_1 = 1110$
$x_2 = 0010$
$x_3 = 0101$
$x_4 = 0110$

$\in (\{0,1\}^{\leq 4})^4$

$S_1 = \{1\}$
$S_2 = \{01, 10\}$
$S_3 = \{001, 011\}$

$\Rightarrow S = \{1, 01, 10, 001, 011\}$

An Offline Algorithm



<u>Algorithm $\mathcal{M}$</u>

Input: queries $q_1,\ldots,q_k$ corresponding to sets $S_1,\ldots,S_k$

1. Let $S = S_1 \cup S_2 \cup \ldots \cup S_k$
2. Replace each x_i with longest $y_i \in S$ which is a prefix of x_i
3. Run your favorite "advanced algorithm" on $(y_1,\ldots,y_n)$

<u>Example:</u>

$x_1 = 1110$
$x_2 = 0010$
$x_3 = 0101$
$x_4 = 0110$

$\in (\{0,1\}^{\leq 4})^4$

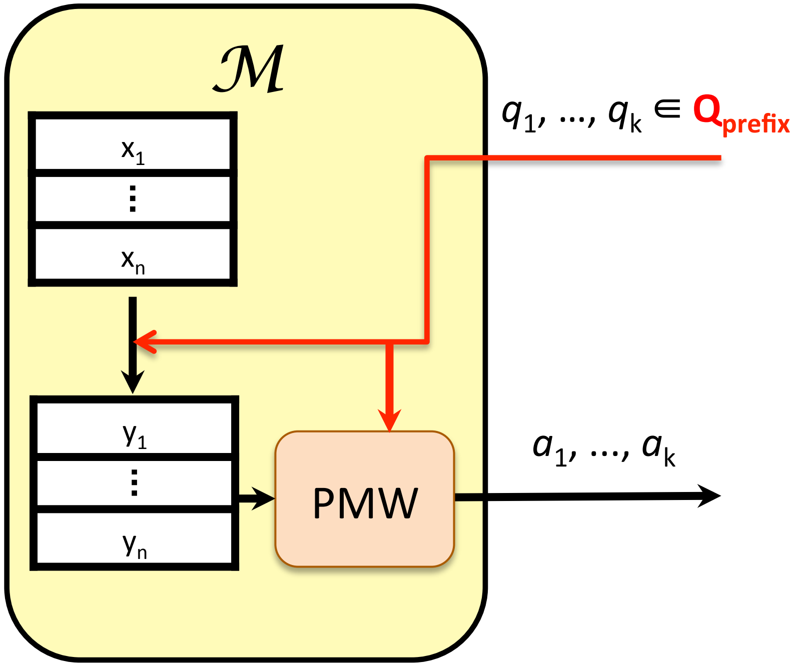
$y_1 = 1$
$y_2 = 001$
$y_3 = 01$
$y_4 = 011$

$\in S^4$

$S_1 = \{1\}$
$S_2 = \{01, 10\}$
$S_3 = \{001, 011\}$

$\Rightarrow S = \{1, 01, 10, 001, 011\}$

An Offline Algorithm



Algorithm $\mathcal{M}$

Input: queries $q_1,\ldots,q_k$ corresponding to sets $S_1,\ldots,S_k$

1. Let $S = S_1 \cup S_2 \cup \ldots \cup S_k$
2. Replace each x_i with longest $y_i \in S$ which is a prefix of x_i
3. Run your favorite "advanced algorithm" on $(y_1,\ldots,y_n)$

Fact 1: All $q_j(y_i) = q_j(x_i)$ (since $z \in S$ is a prefix of x_i iff z is a prefix of y_i)

Fact 2: y_i's come from a universe of size only kd (i.e. dimension $\log(kd)$)

$\Rightarrow$ Private Mult. Weights can answer $k = \exp(\Omega(n/\log^{1/2}(kd)))$ queries

For $d = \mathrm{poly}(n)$, solve to get $k = \exp(\Omega(n^{1/2}))$

Our Results

All three models are distinct

- **Offline ≠ Online**

  Family $\mathbf{Q}_{\text{prefix}}$ of counting queries

  Offline: Can answer $k = \exp(\Omega(n^{1/2}))$ queries

  Online: Can only answer $k = O(n^2)$ queries

- **Online ≠ Adaptive**

  Family $\mathbf{Q}_{\text{corr}}$ of "search" queries

Online: $k = \exp(\Omega(n))$ queries

Adaptive: $k = O(1)$ queries

An Online Lower Bound

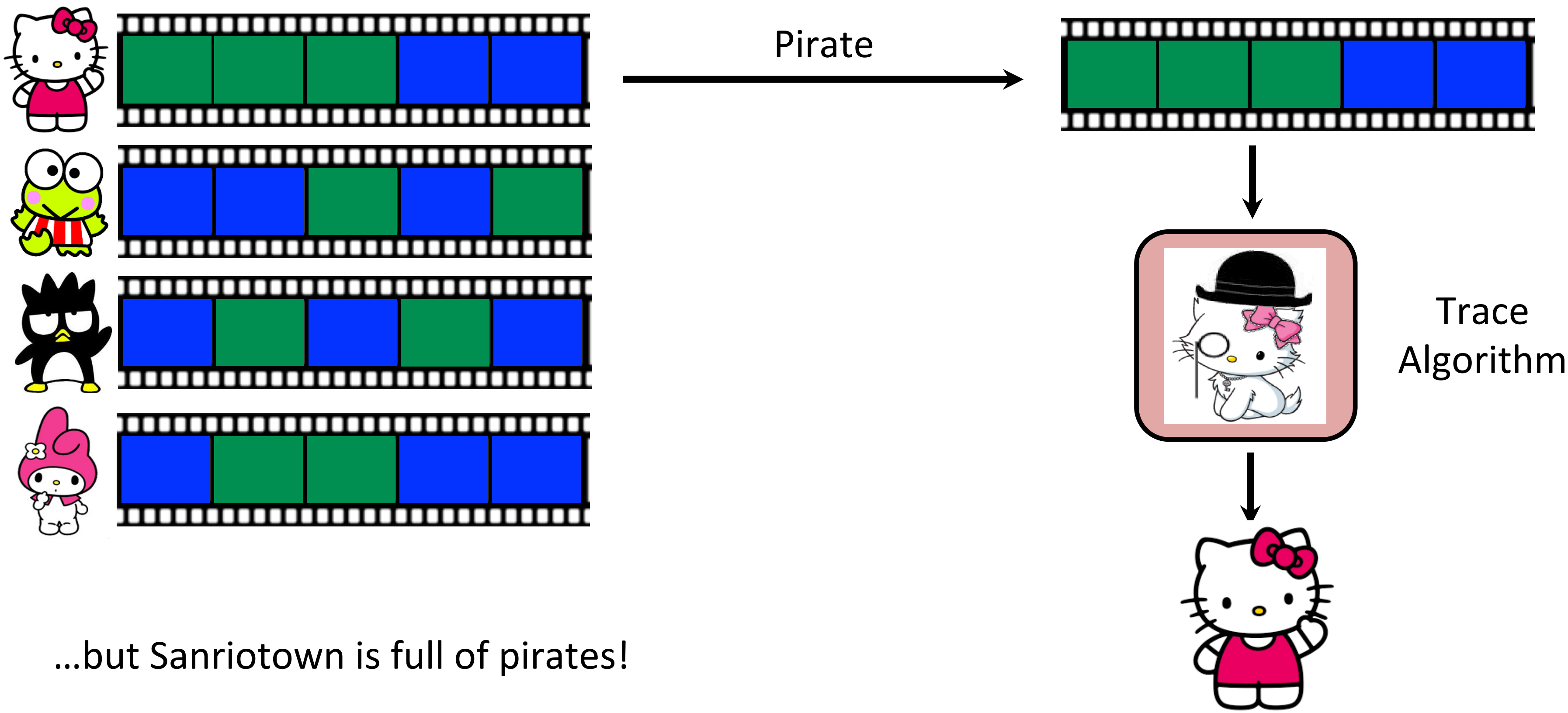
- Lower bound for attribute means via fingerprinting codes [B.-Ullman-Vadhan14]

- "Embed" attribute means into online prefix queries

  See: [Bassily-Smith-Thakurta15, Dwork-Talwar-Thakurta-Zhang15, Steinke-Ullman15, B.-Nissim-Stemmer16]

Fingerprinting Codes [Boneh-Shaw95]

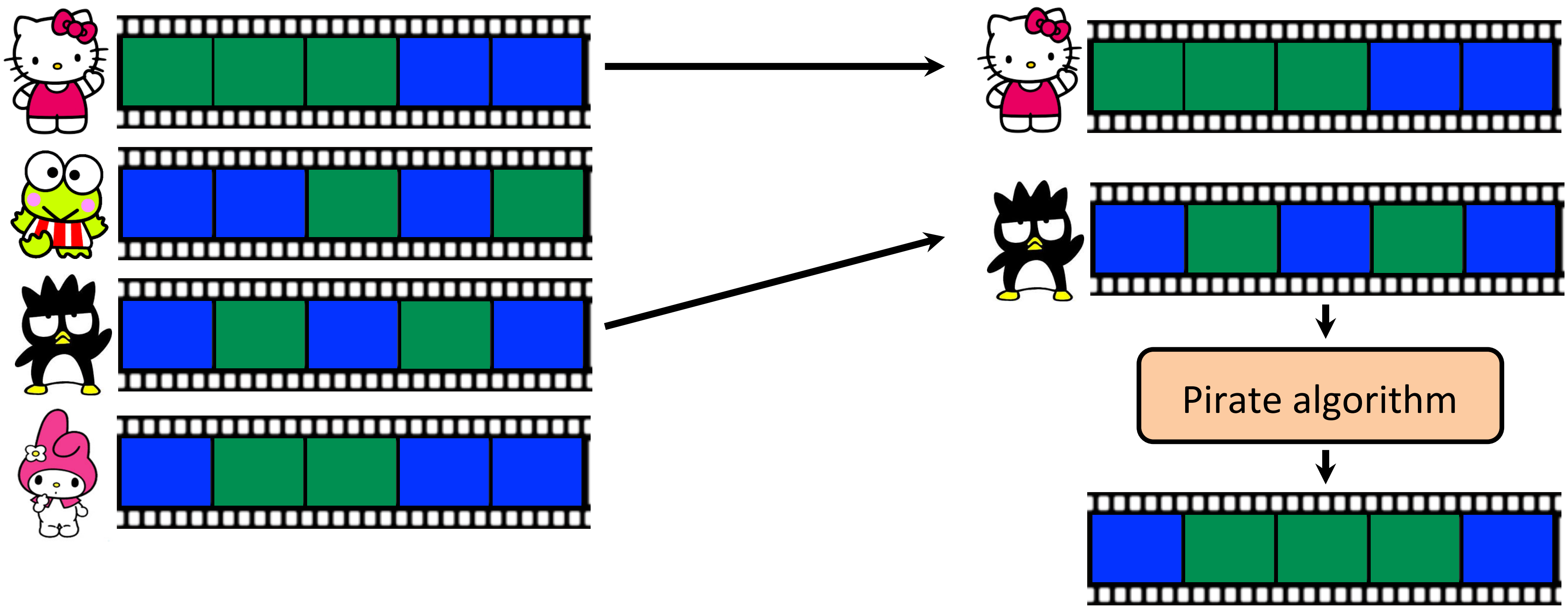
I want to distribute my new movie



...but Sanriotown is full of pirates!

Fingerprinting Codes [Boneh-Shaw95]

I want to distribute my new movie



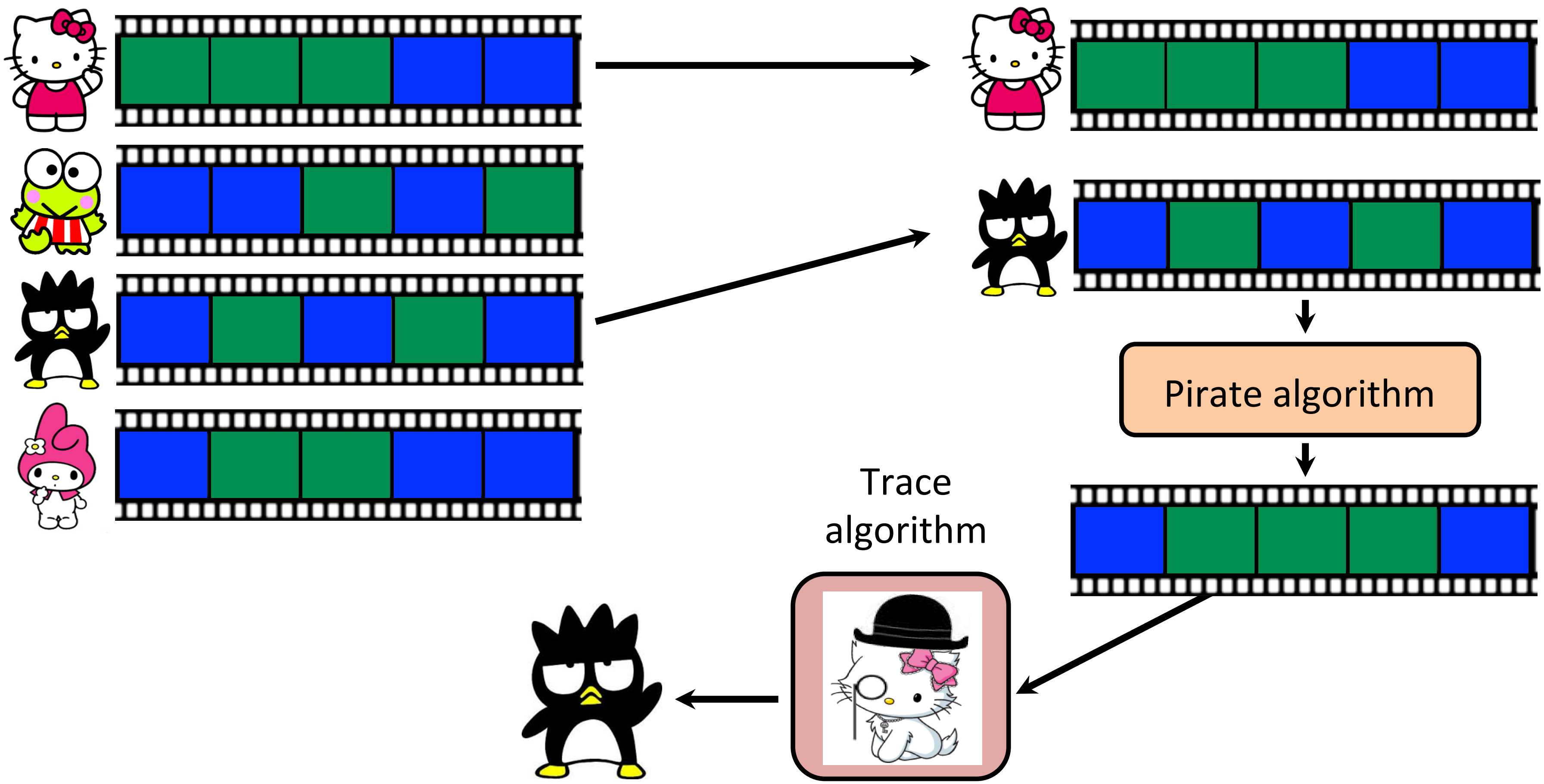
…but Sanriotown is full of pirates!

Who collude against me!

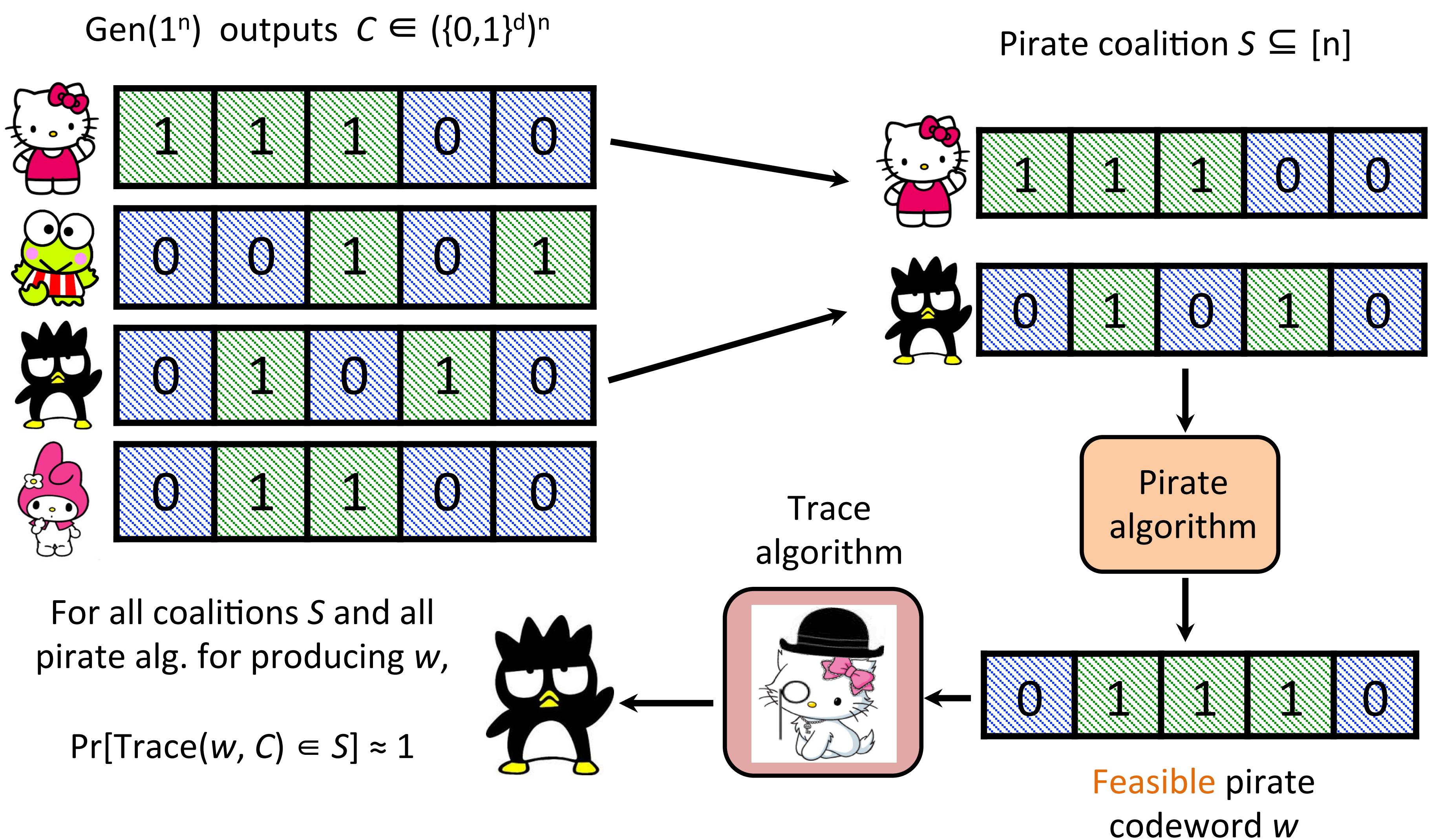
Fingerprinting Codes [Boneh-Shaw95]

I want to distribute my new movie

Hello Kitty's Gradient Descent



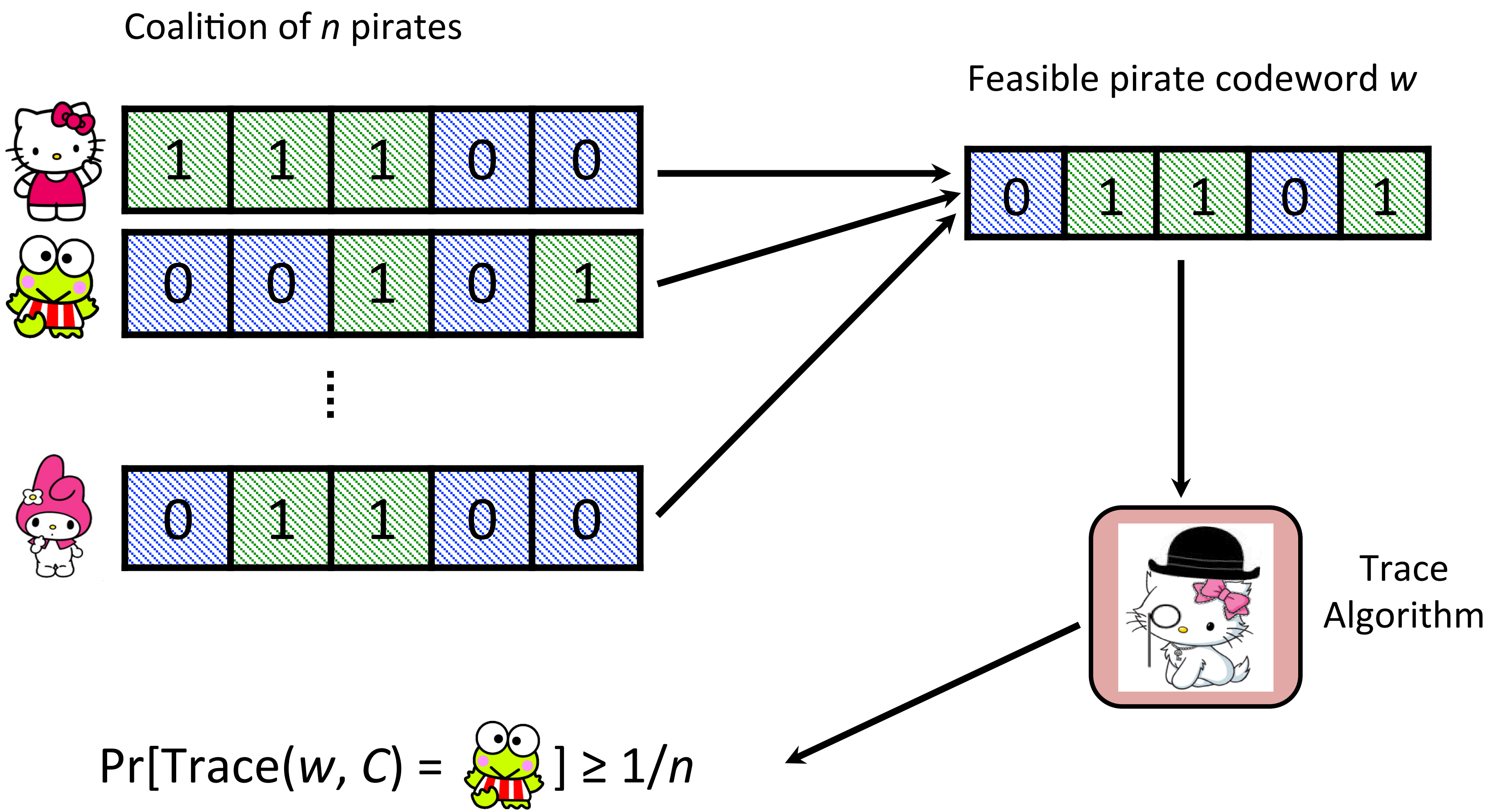
Fingerprinting Codes [Boneh-Shaw95]



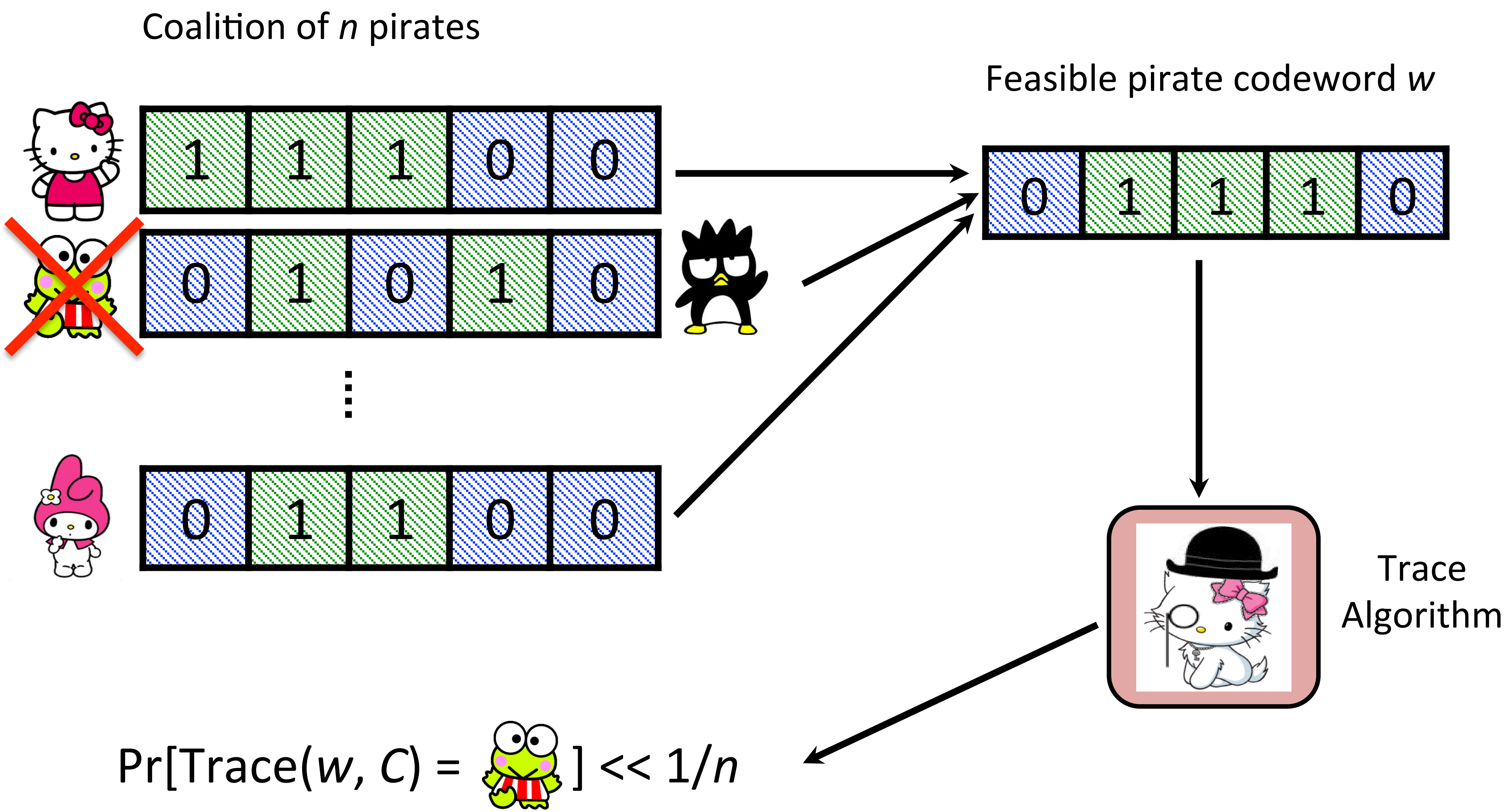
For all coalitions S and all pirate alg. for producing w,

$\Pr[\mathrm{Trace}(w, C) \in S] \approx 1$

FP Codes vs. Diff. Privacy



FP Codes vs. Diff. Privacy

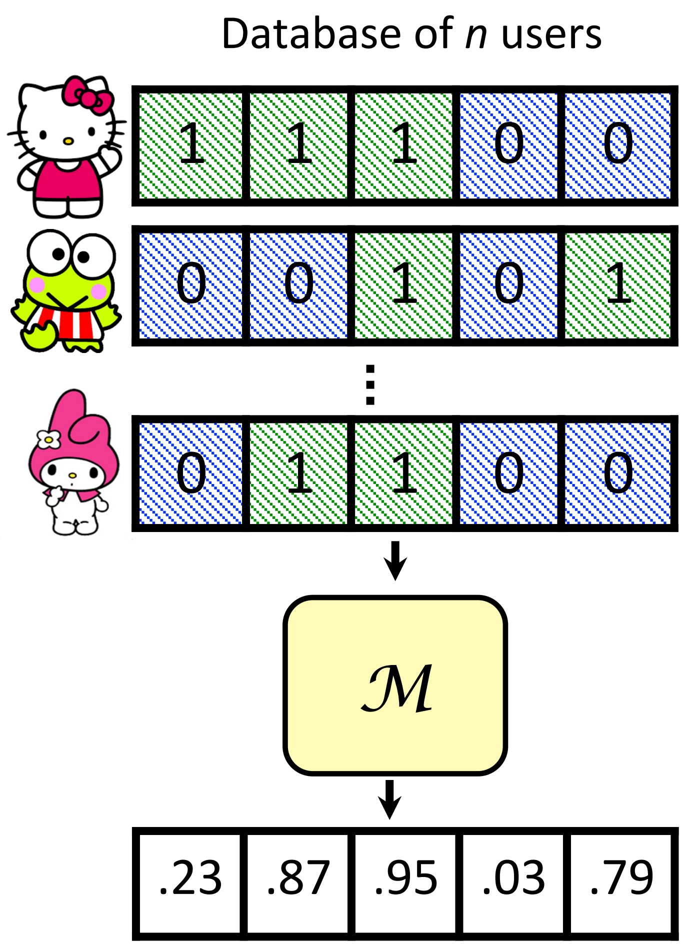


FP Codes vs. Diff. Privacy

Trace behaves very differently depending on whether  is in the coalition

Fingerprinting codes are the "opposite" of differential privacy!

Lower Bound for Attribute Means

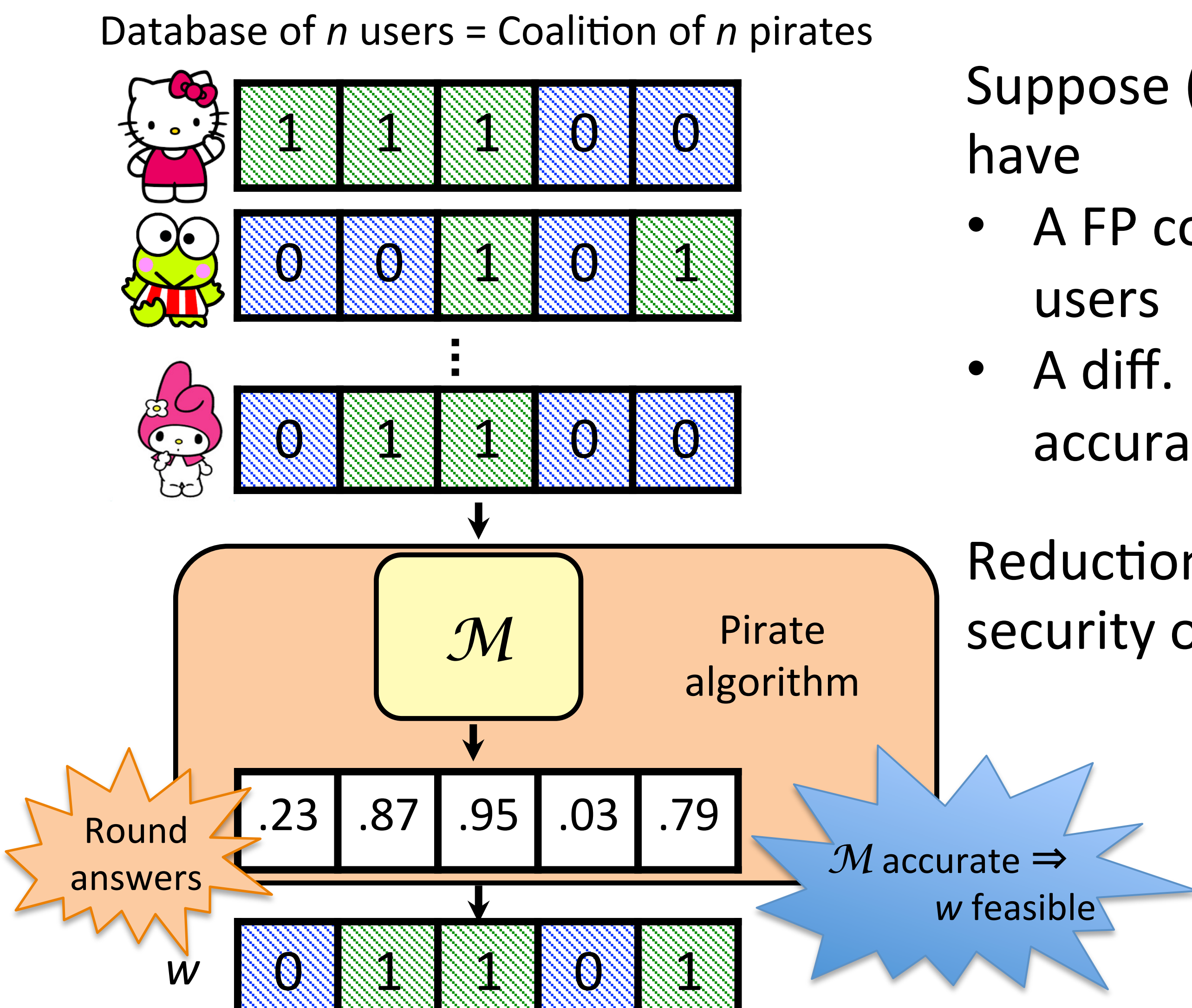


Suppose (for contradiction) we have

- A FP code of length k for (n+1) users
- A diff. private $\mathcal{M}$ that is accurate for k attribute means

Reduction: Use $\mathcal{M}$ to break security of the FP code

Lower Bound for Attribute Means

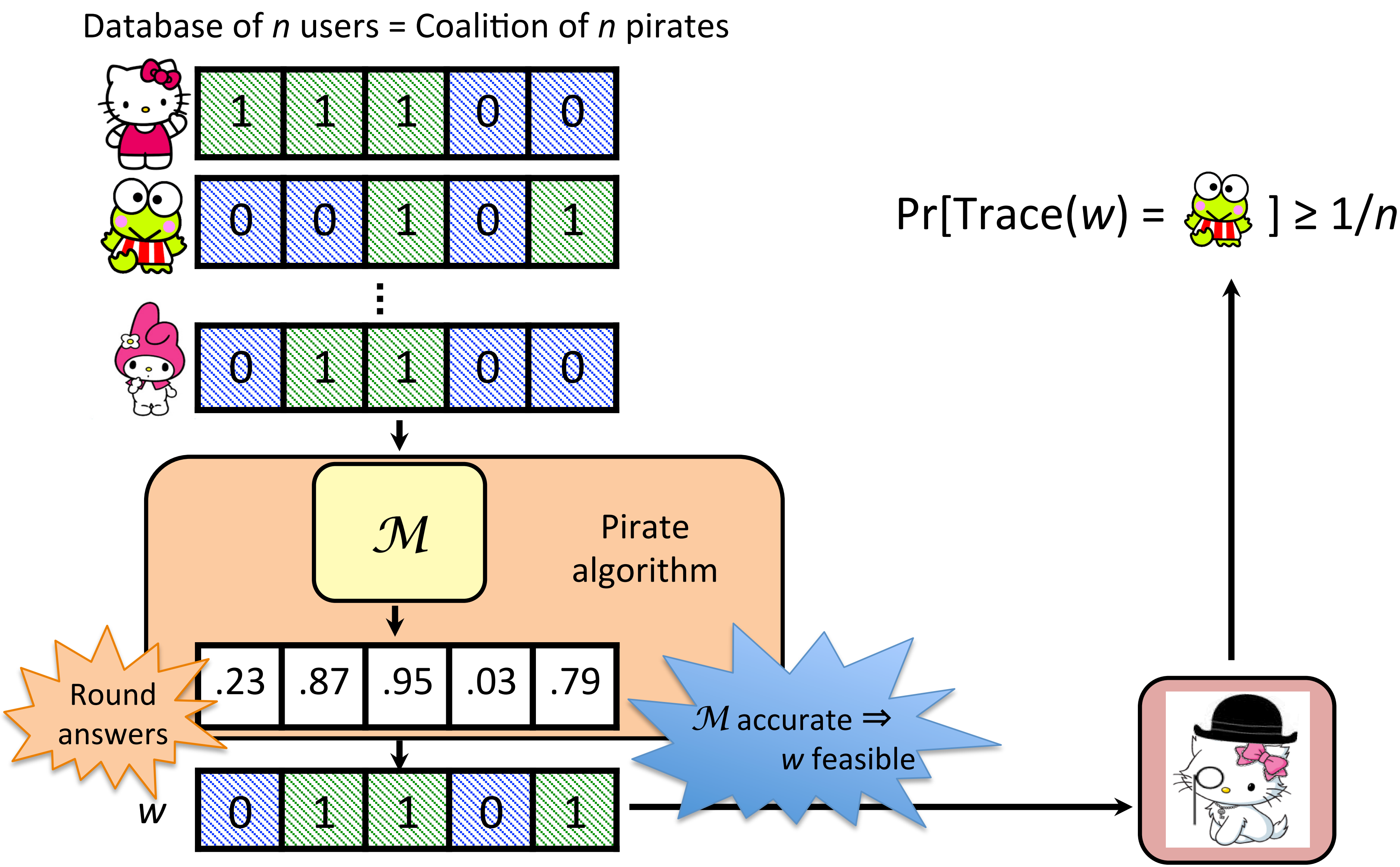


Suppose (for contradiction) we have

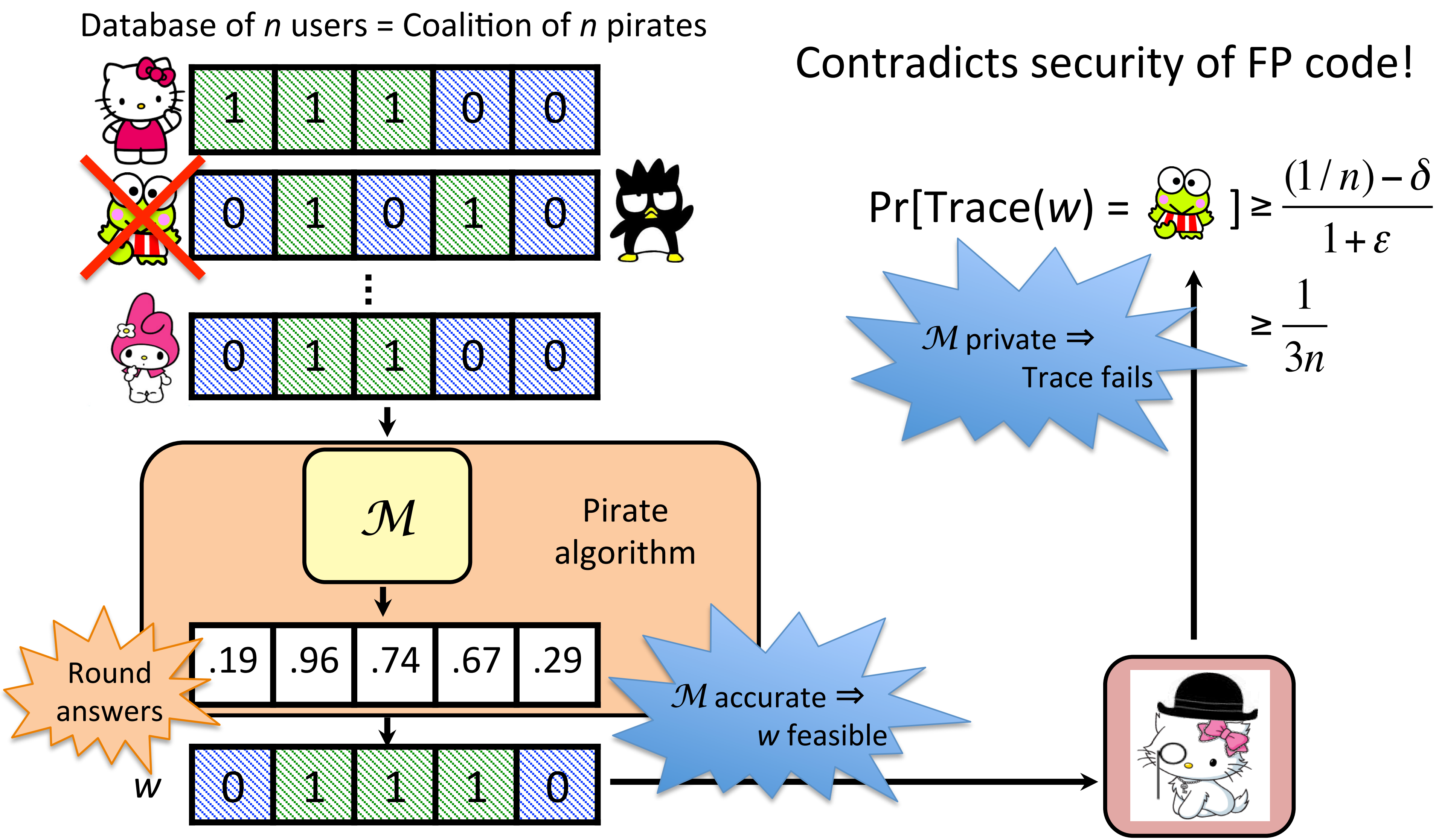
- A FP code of length *k* for (*n*+1) users
- A diff. private $\mathcal{M}$ that is accurate for *k* attribute means

Reduction: Use $\mathcal{M}$ to break security of the FP code

Lower Bound for Attribute Means



Lower Bound for Attribute Means



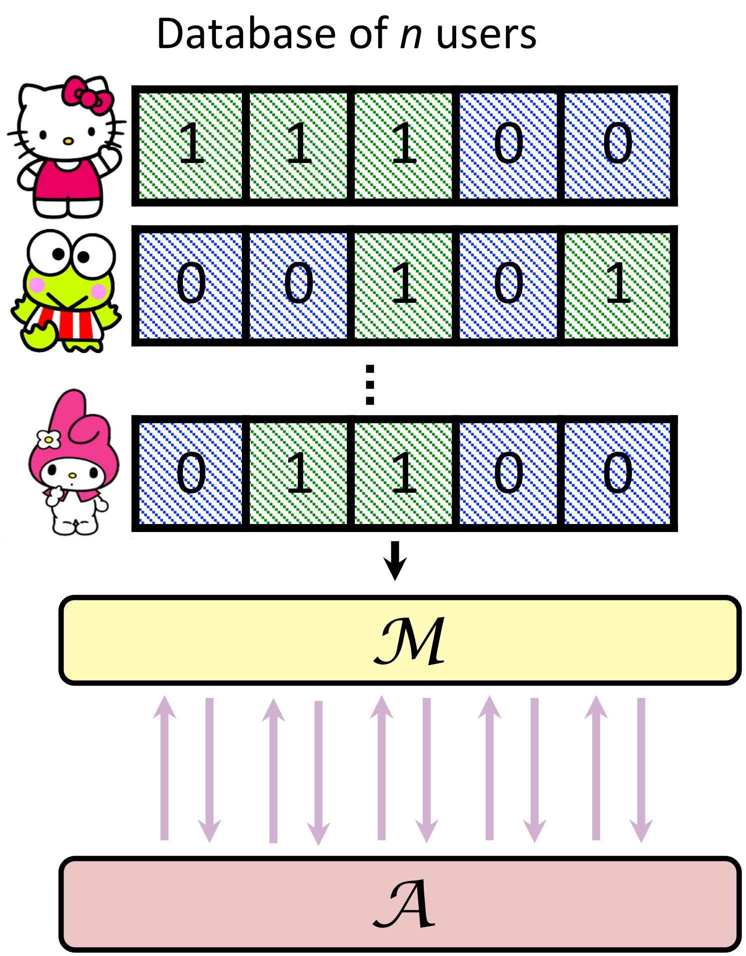
Lower Bound for Attribute Means

- ∃ FP code for $\boldsymbol{n}$ users with length $\boldsymbol{k}$
  $\Rightarrow$ $\boldsymbol{n}$ samples enables < $\boldsymbol{k}$ attribute means

- [Tardos03] ∃ FP code for $\boldsymbol{n}$ users of length $\boldsymbol{k} = O(\boldsymbol{n}^2)$
  $\therefore$ attribute means require $\boldsymbol{k} \leq O(\boldsymbol{n}^2)$

Next: How to embed attribute means into online
prefix queries

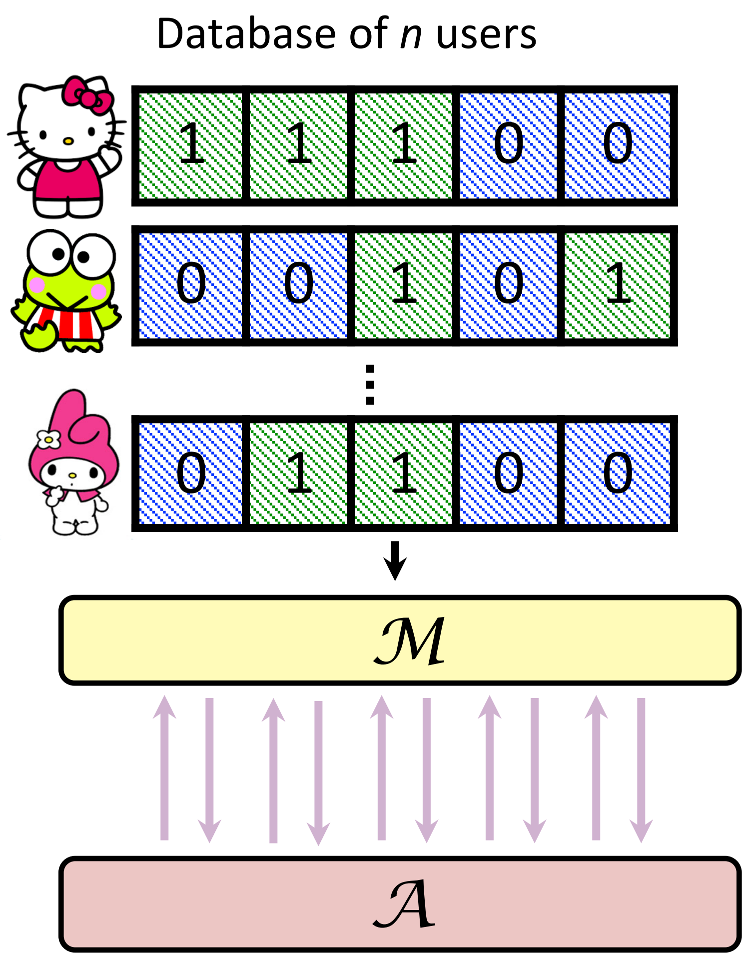
Lower Bound for Prefix Queries



Suppose $\mathcal{M}$ can answer k prefix queries presented online

Reduction: Use $\mathcal{M}$ to answer k attribute mean queries

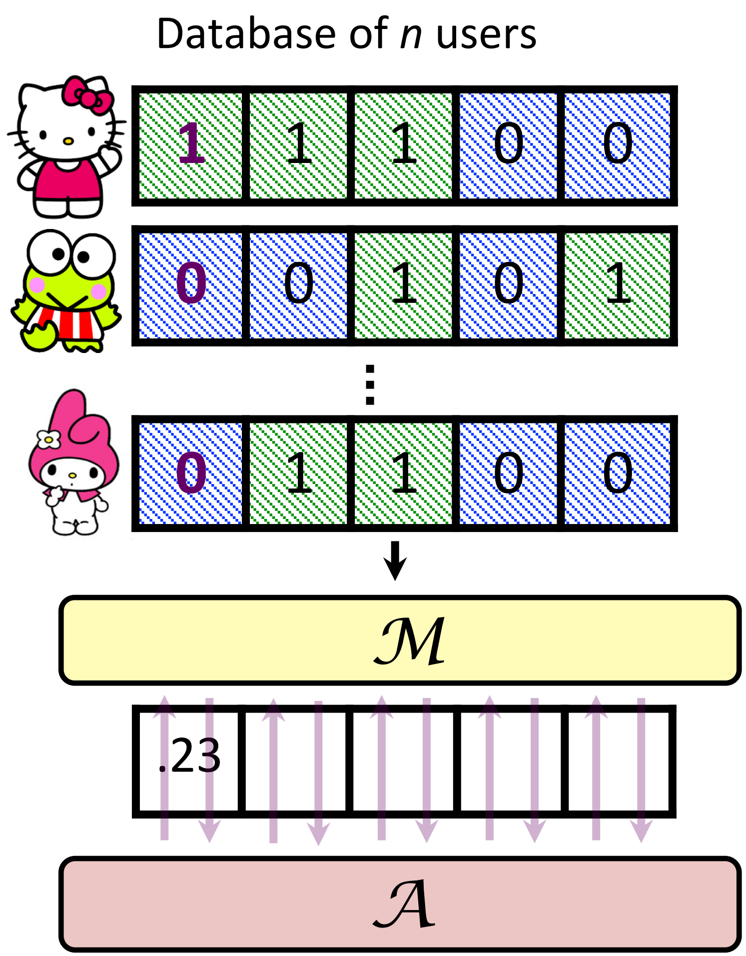
Lower Bound for Prefix Queries



<u>Queries:</u>

Recall $q_S(x) = 1$ iff $\exists y \in S$ that is a prefix of x

Lower Bound for Prefix Queries

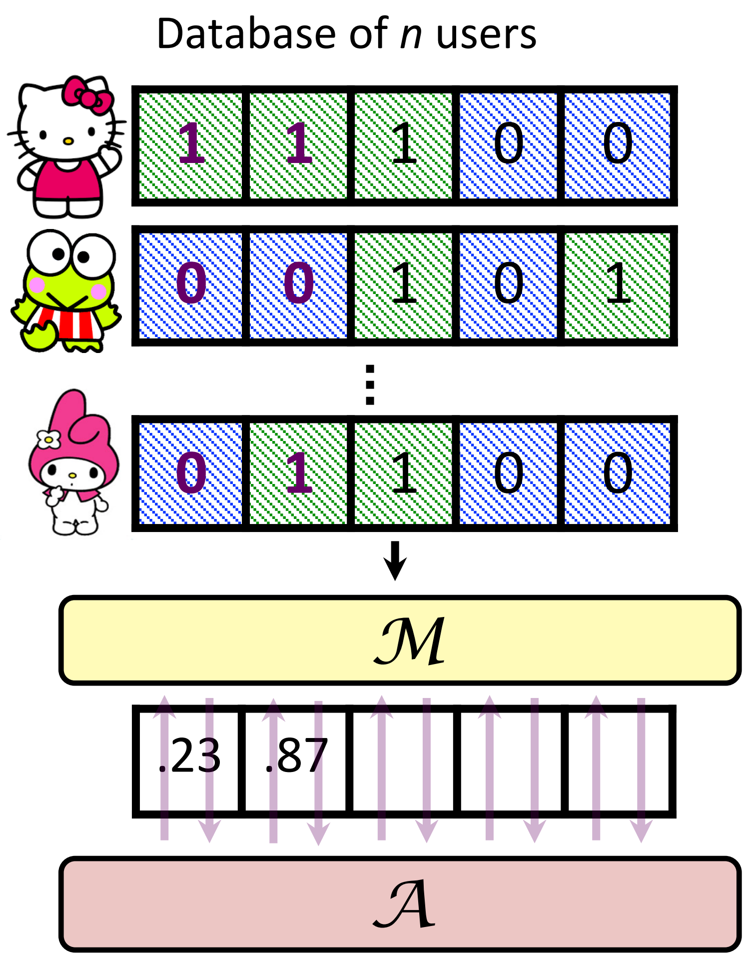


Queries:

Recall $q_S(x) = 1$ iff $\exists y \in S$ that is a prefix of x

$S_1 = \{1, 1, \ldots, 1\}$

Lower Bound for Prefix Queries



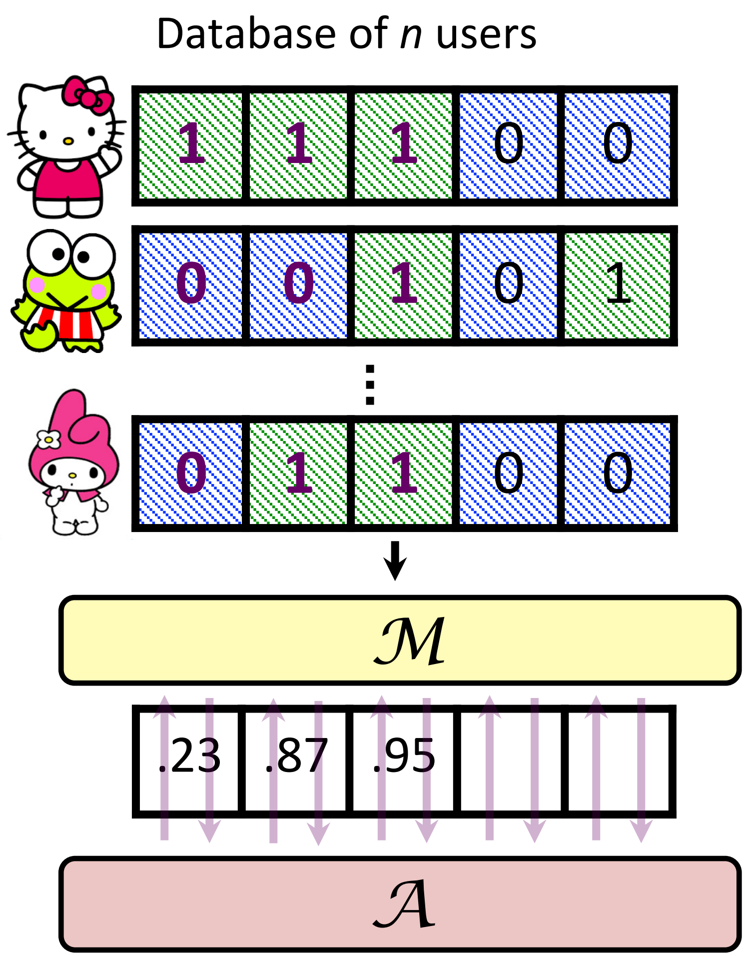
<u>Queries:</u>

Recall $q_S(x) = 1$ iff $\exists y \in S$ that is a prefix of x

$S_1 = \{1, 1, \ldots, 1\}$

$S_2 = \{11, 01, \ldots, 01\}$

Lower Bound for Prefix Queries



<u>Queries:</u>

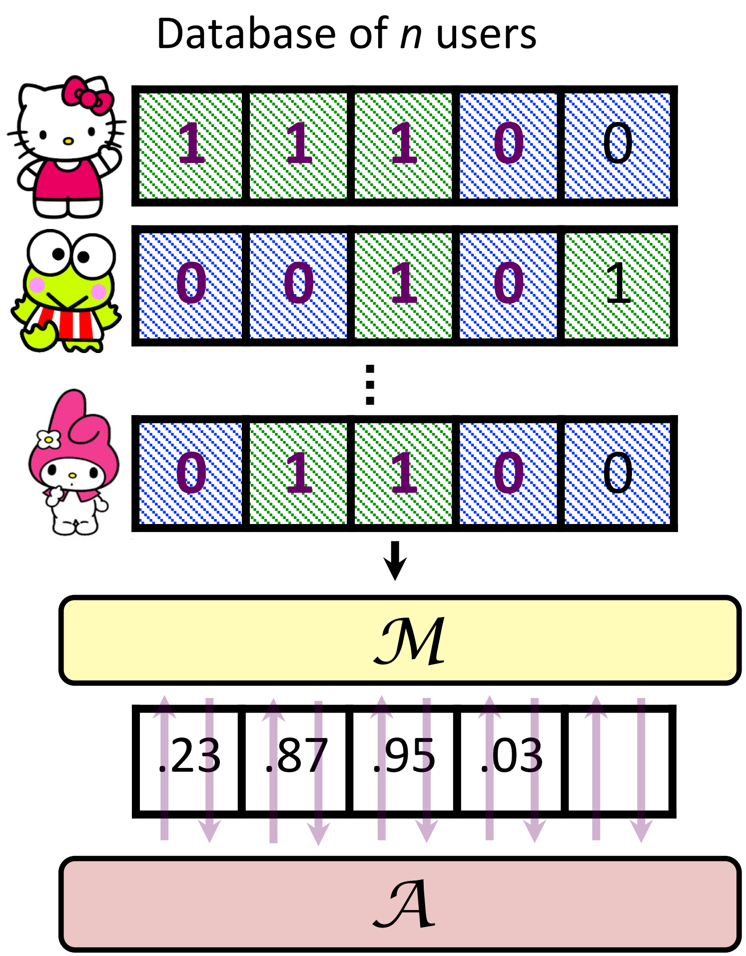
Recall $q_S(x) = 1$ iff $\exists y \in S$ that is a prefix of x

$S_1 = \{1, 1, \ldots, 1\}$

$S_2 = \{11, 01, \ldots, 01\}$

$S_3 = \{111, 001, \ldots, 011\}$

Lower Bound for Prefix Queries



<u>Queries:</u>

Recall $q_S(x) = 1$ iff $\exists y \in S$ that is a prefix of x

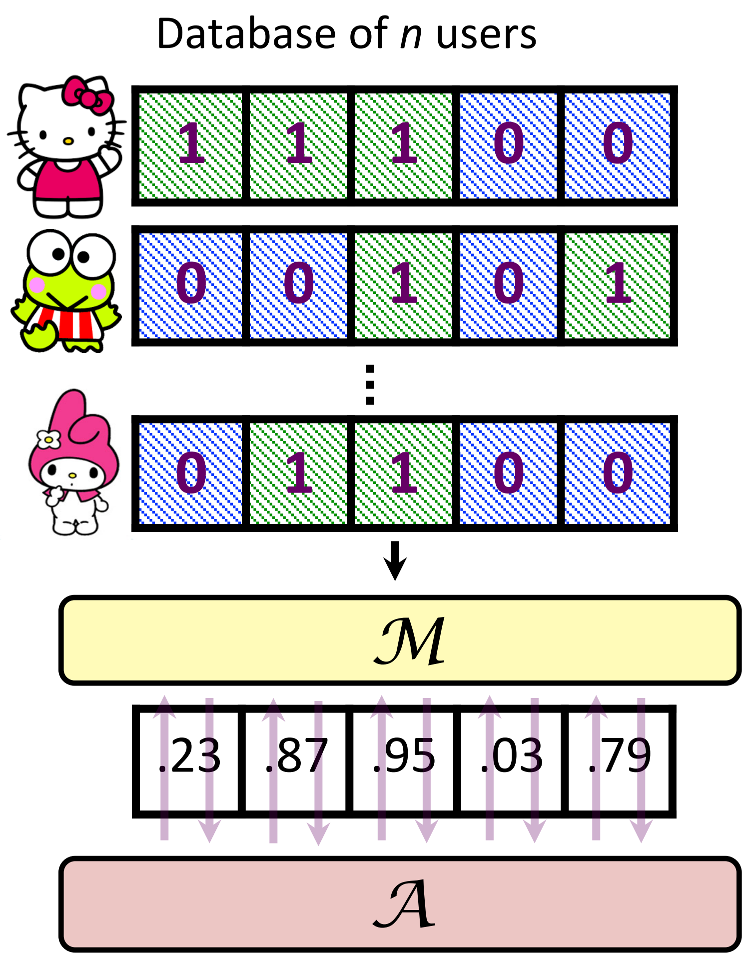
$S_1 = \{1, 1, \ldots, 1\}$

$S_2 = \{11, 01, \ldots, 01\}$

$S_3 = \{111, 001, \ldots, 011\}$

$S_4 = \{1111, 0011, \ldots, 0111\}$

Lower Bound for Prefix Queries

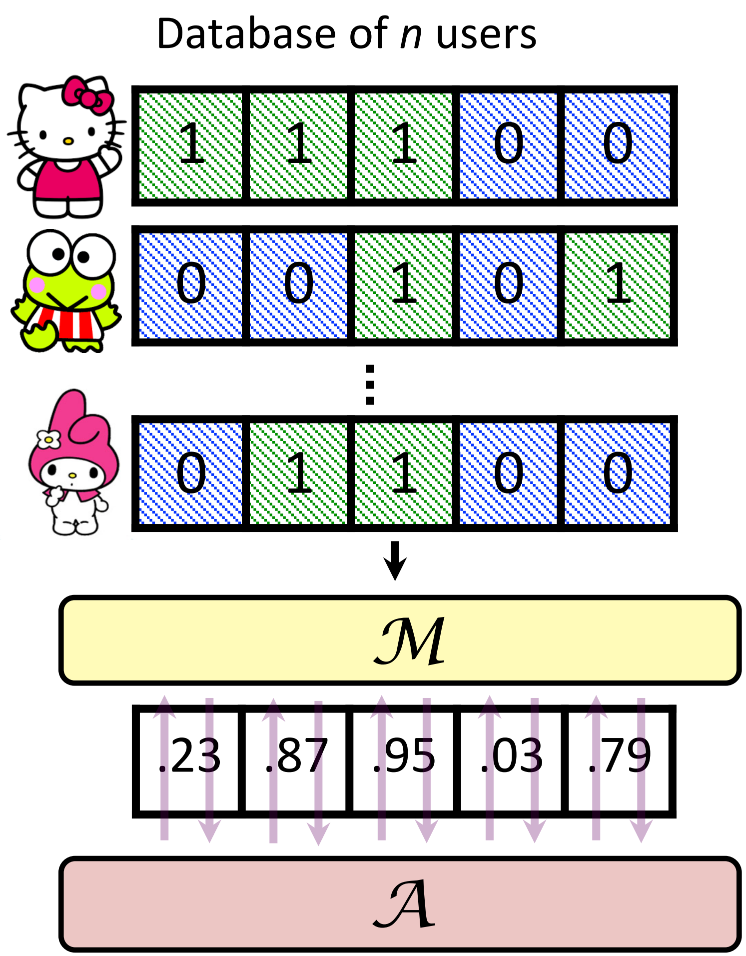


<u>Queries:</u>

Recall $q_S(x) = 1$ iff $\exists y \in S$ that is a prefix of x

$S_1 = \{1, 1, \ldots, 1\}$
$S_2 = \{11, 01, \ldots, 01\}$
$S_3 = \{111, 001, \ldots, 011\}$
$S_4 = \{1111, 0011, \ldots, 0111\}$
$S_5 = \{11101, 00101, \ldots, 01101\}$

Lower Bound for Prefix Queries



<u>Queries:</u>

Recall $q_S(x) = 1$ iff $\exists y \in S$ that is a prefix of x

$S_1 = \{1, 1, \ldots, 1\}$
$S_2 = \{C_{1,1}1, \ldots, C_{n+1,1}1\}$
$S_3 = \{C_{1,1}C_{1,2}1, \ldots, C_{n+1,1}C_{n+1,2}1\}$
$S_4 = \{C_{1,1}C_{1,2}C_{1,3}1, \ldots\}$
…

<u>Fact 1:</u> $q_j(D) = j^{th}$ attribute mean

<u>Fact 2:</u> $q_1, \ldots, q_{j-1}$ reveal nothing about q_j

(But q_j reveals answers to $q_1, \ldots, q_{j-1}$!)

Lower Bound for Prefix Queries

- $\boldsymbol{n}$ samples suffice for $\boldsymbol{k}$ online prefix queries
  $\Rightarrow$ $\boldsymbol{n}$ samples suffice for $\boldsymbol{k}$ attribute means*

- Attribute mean lower bound $\boldsymbol{k} = O(\boldsymbol{n}^2)$
  $\therefore$ online prefix queries require $\boldsymbol{k} \leq O(\boldsymbol{n}^2)$
  (Even for $\boldsymbol{d} = O(\boldsymbol{n}^2)$)

*Not quite black-box use of FPCs / attribute mean lower bound, but follows from FP code analysis of [Steinke-Ullman15, Dwork-Smith-Steinke-Ullman-Vadhan15]

Our Results

All three models are distinct

- **Offline ≠ Online**

  Family $\mathbf{Q}_{\mathbf{prefix}}$ of counting queries

  Offline: Can answer $\boldsymbol{k} = \exp(\Omega(\boldsymbol{n}^{1/2}))$ queries

  Online: Can only answer $\boldsymbol{k} = O(\boldsymbol{n}^2)$ queries

- **Online ≠ Adaptive**

  Family $\mathbf{Q}_{\mathbf{corr}}$ of "search" queries

  Online: $\boldsymbol{k} = \exp(\Omega(\boldsymbol{n}))$ queries

  Adaptive: $\boldsymbol{k} = O(1)$ queries

Online vs. Adaptive (Idea)

$\mathbf{Q_{corr}} = \{\ q_S : \{0,1\}^n \rightarrow \{0,1\}\ \}$ ***Not counting queries***

For $S = \{y_1,\ldots,y_m \in \{0,1\}^n\}$ and $x \in \{0,1\}^n$:

"Find me a vector $z \in \{0,1\}^n$ that is highly correlated with x, but not too correlated with any y_j"

Intuition

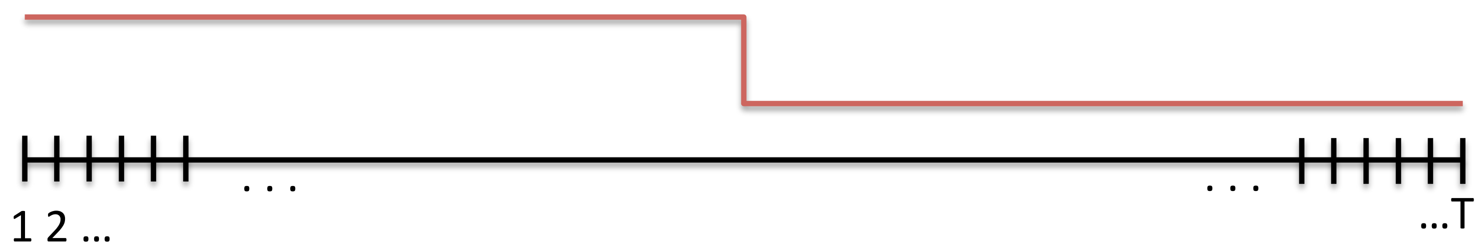
Online: Randomized response [Warner65] – Choose z once and for all with $z_i = \mathrm{Round}(x_i + \mathrm{Noise}(1/\varepsilon))$

Adaptive: Picking queries strategically enables a reconstruction attack

Conclusions

- To answer many queries with differential privacy, it can help to "make up your mind"
- Open questions:
  - Can counting queries separate online vs. adaptive?
  - Are there *natural* tasks that separate these models?

    Some evidence for one-dimensional thresholds



Thank you!