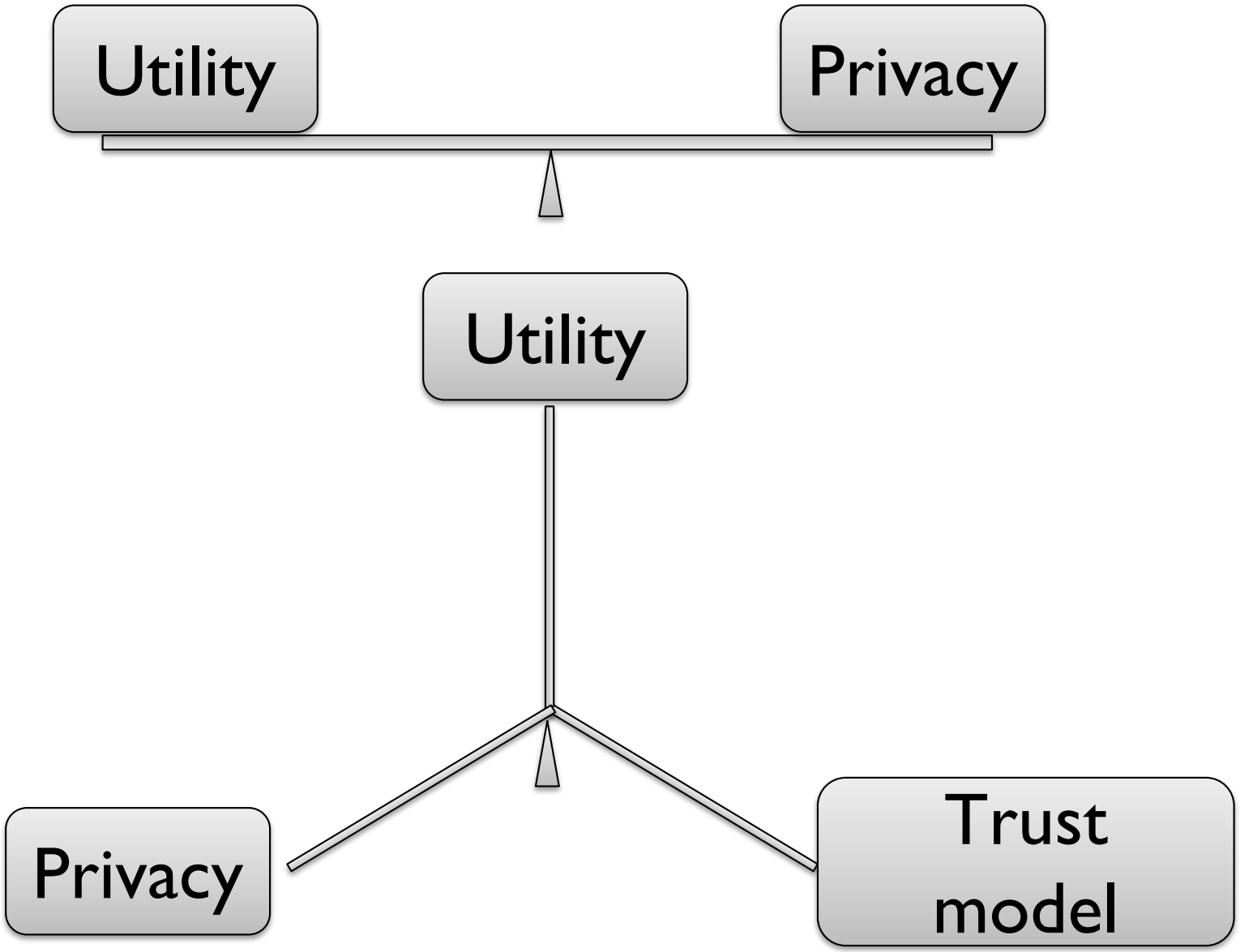
BU CS599
Spring 2025

Lecture 24: Distributed Models

Jonathan Ullman
NEU

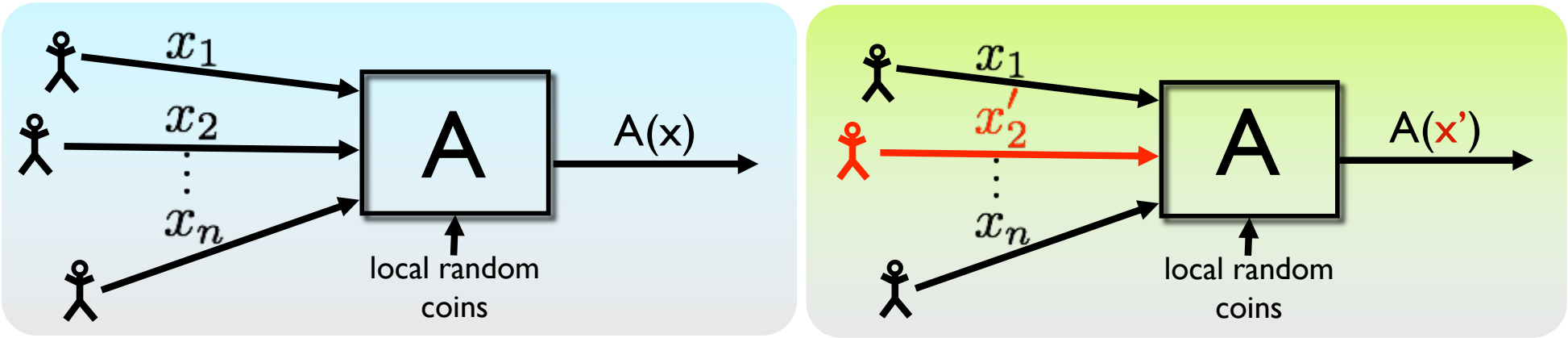
Adam Smith
BU


Utility
Privacy
Utility
Privacy
Trust
model

Distributed Models

- Local Differential Privacy
  - Randomized Response Strikes Back
  - Limitations of the Model
- Cryptographic Tools
  - Encryption
  - Multiparty Computation
- What's next?
  - Efficient "federated" protocols?
  - Minimal crypto primitives?

Differential Privacy



x' is a neighbor of x
if they differ in one data point

Neighboring databases induce **close** distributions on outputs

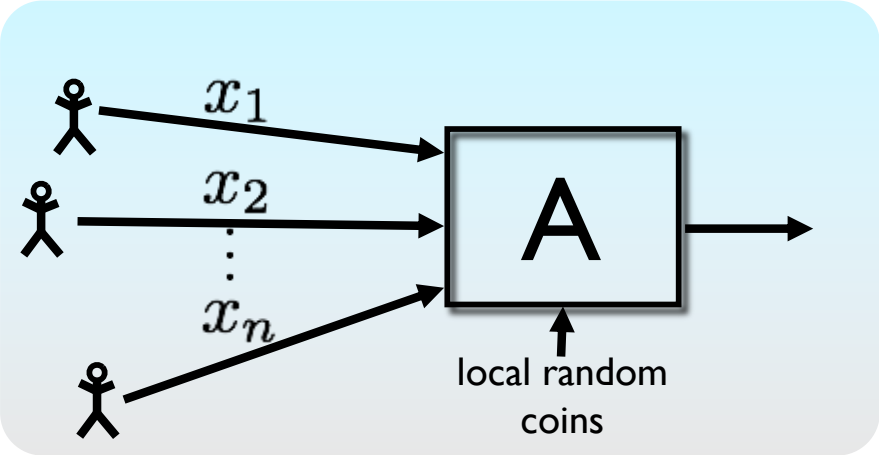
Definition: A is ε-differentially private if,

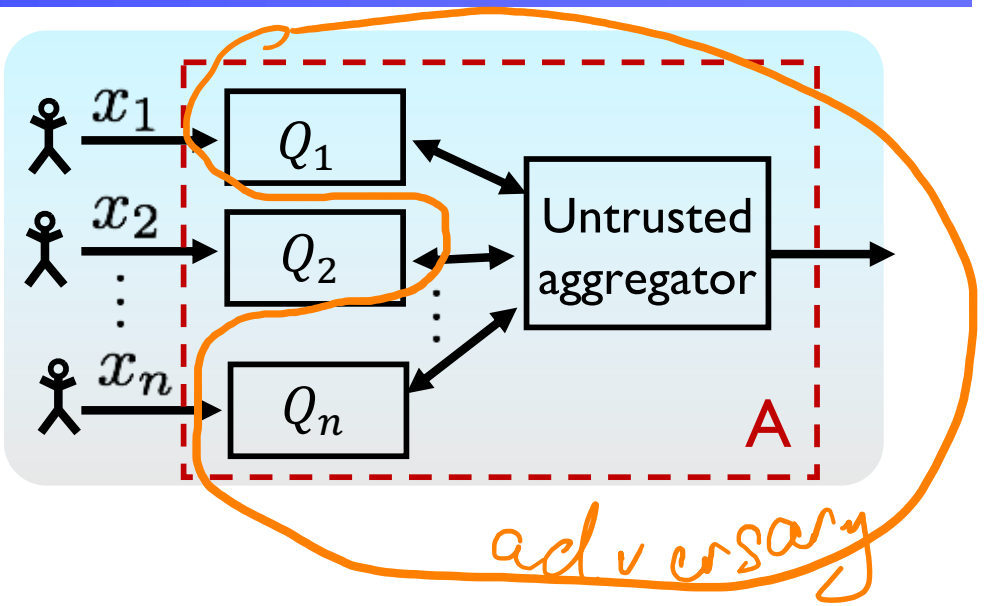
for all neighbors x, x',

for all sets of outputs T

$$\Pr_{\text{coins of } A}(A(x) \in T) \leq e^{\varepsilon} \cdot \Pr_{\text{coins of } A}(A(x') \in T)$$

Local Model for Privacy





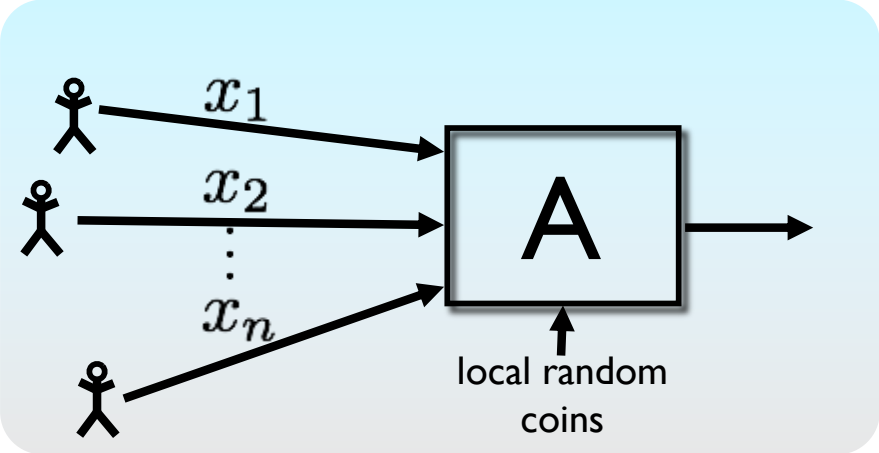
- "Local" model
  - Person i randomizes their own data
  - Attacker sees everything except player i's local state

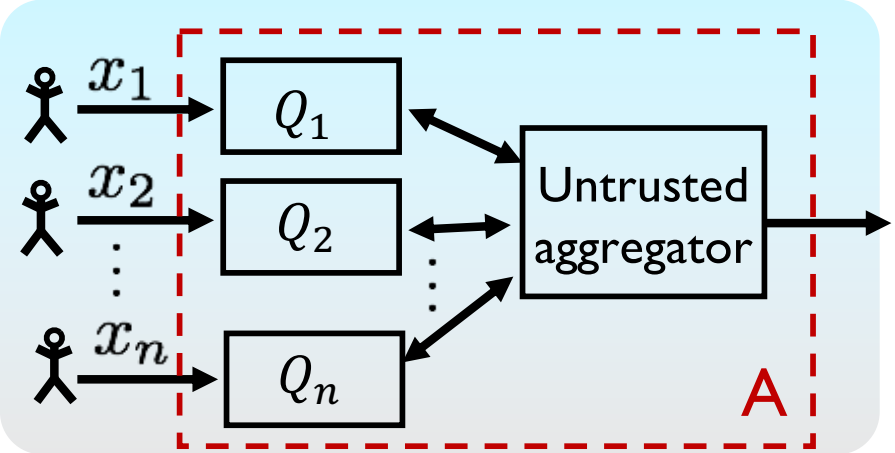
- Definition: A is ε-locally differentially private if for all i:
  - for all neighbors $\boldsymbol{x}$, $\boldsymbol{x'}$ that differ in position i
  - for all local coins $\boldsymbol{r_{-i}}$ of all other parties,
  - for all transcripts t:

$$\Pr_{\text{coins } r_i}(A(\boldsymbol{x}, r_{-i}) = t) \leq e^{\varepsilon} \cdot \Pr_{\text{coins } r_i}(A(\boldsymbol{x'}, r_{-i}) = t)$$

Non-interactive
$\delta = 0$ w.l.o.g.
1 message each
[Duchi-Nelson-Steinhardt 18]
General model
Many messages

Local Model for Privacy





- Pros
  - No trusted curator
  - No single point of failure
  - Highly distributed
  - Beautiful algorithms
- Cons
  - Lower accuracy
    - Proportions: $\Theta\left(\frac{1}{\varepsilon\sqrt{n}}\right)$ error [Beimel-Nissim-Omri'08,Chan-Shi-Song'12,Duchi-Jordan-Wainwright'13, Joseph-Mao-Neel-Roth'19] vs $O\left(\frac{1}{n\varepsilon}\right)$ central
    - Selection: $\Theta\left(\frac{1}{\varepsilon}\sqrt{\frac{d}{n}}\right)$ error [DJW13, Ullman17] vs $\Theta(\frac{\log d}{n\varepsilon})$ central [exp. mechanism]
  - Correctness requires honesty (e.g. [Cheu-Smith-Ullman '21])

Reminder: Randomized response

- Each person has data $x_i \in \mathcal{X}$
  - Analyst wants to know sum of $\varphi: \mathcal{X} \to \{0,1\}$ over $\boldsymbol{x}$
- Randomization operator takes $z \in \{0,1\}$:

$$R(z) = \begin{cases} z & w.p. \ \frac{e^{\varepsilon}}{e^{\varepsilon}+1} \\ 1-z & w.p. \ \frac{1}{e^{\varepsilon}+1} \end{cases}$$

- How can we estimate a proportion?
  - $A(x_1, \ldots, x_n)$:
    - For each i, let $Y_i = R(\varphi(x_i))$
    - Return $A = \sum_i (aY_i - b)$
  - Set $a = \frac{e^{\varepsilon}+1}{e^{\varepsilon}-1}, b = \frac{1}{e^{\varepsilon}-1}$ to get $\mathbb{E}(A) = \sum_i \varphi(x_i)$
- **Proposition:** $\sqrt{\mathbb{E}\left(A - \sum_i \varphi(x_i)\right)^2} \leq \frac{e^{\varepsilon/2}}{e^{\varepsilon}-1}\sqrt{n}$. $\approx \frac{\sqrt{n}}{\varepsilon}$ when ε small

Randomized response is optimal

- **Theorem**: Every LDP algorithm has worst-case error $\Omega(\frac{1}{\varepsilon\sqrt{n}})$ for estimating proportion of 1's.
  - ➢ Cleanest proof via mutual information argument
- **Simpler theorem**: Every noninteractive LDP algorithm with $\varepsilon \leq 1$ has worst-case error $\Omega\left(\frac{1}{\sqrt{n}}\right)$.

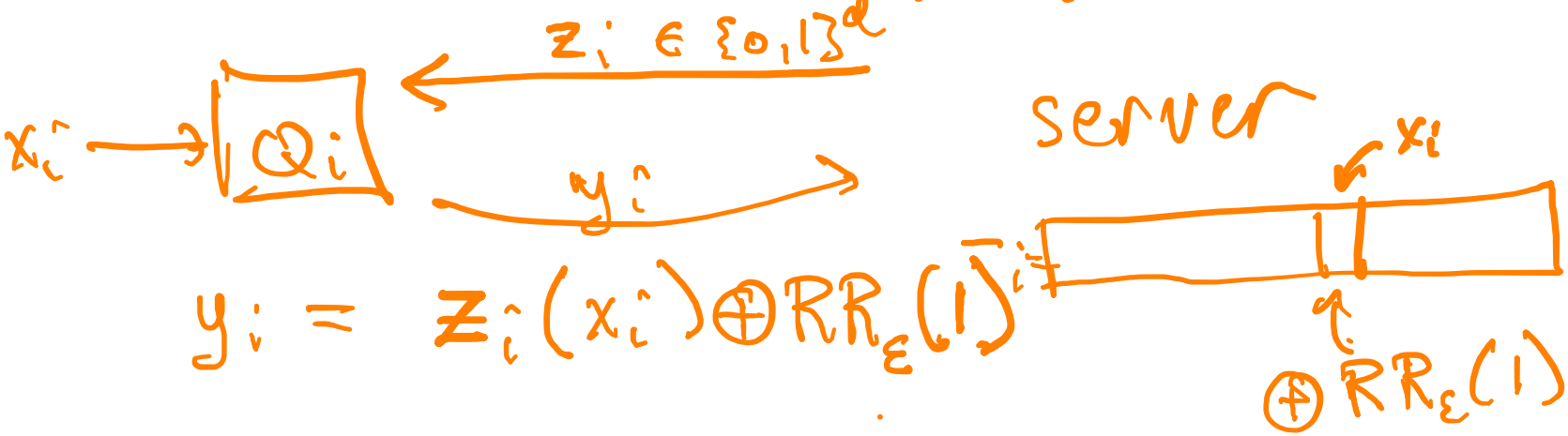
- Fix randomizers $Q_1, \ldots, Q_n$.
- Pick $X_1 \ldots X_n \sim iid \{0,1\}$ (uniform)
- Conditioned on $Y_1, \ldots, Y_n$ $(Y_i = Q_i(X_i))$
  1) X_i are still independent!
  2) $\Pr(X_i = 0 \mid Y_i = y_i) = (\text{Bayes' rule}) \geq \frac{1}{e^{\varepsilon}+1}$
  3) conditional distrib on $\frac{1}{n}\sum X_i$ has variance $\Omega(1/n) \geq 1/3$

Case Study: Histograms/Heavy Hitters

- Inputs: $x_1, \ldots, x_n \in [d]$
- Goal: Find $n_1, n_2, \ldots, n_d \in \mathbb{N}$, where $n_j = \#\{i : x_i = j\}$
- How can use RR?
  1. Randomized the input directly:
     a) Write each x_i as string in $\{0,1\}^{\log d}$
     b) Apply $RR_{\varepsilon'}$ to each bit (for $\varepsilon' \approx \varepsilon/\sqrt{\log d}$)
  2. Randomize the one-hot encoding of x_i
     a) Write $x_i \in [d]$ as $(0,0, \ldots, 0,1,0, \ldots, 0)$ with 1 in position x_i
     b) Homework 1, Problem 3: Can apply $RR_{\varepsilon/2}$ to each bit.
     c) Estimate frequency of all items with error $O\left(\frac{1}{\varepsilon}\sqrt{\frac{\ln d}{n}}\right)$
     d) Drawbacks?
        (Communication)

Compressing the communication

(1) Move comm to server/aggregator



$y_i = z_i(x_i) \oplus RR_\varepsilon(1)^d$

Server stores $\tilde{z}_i = y_i \oplus z_i$

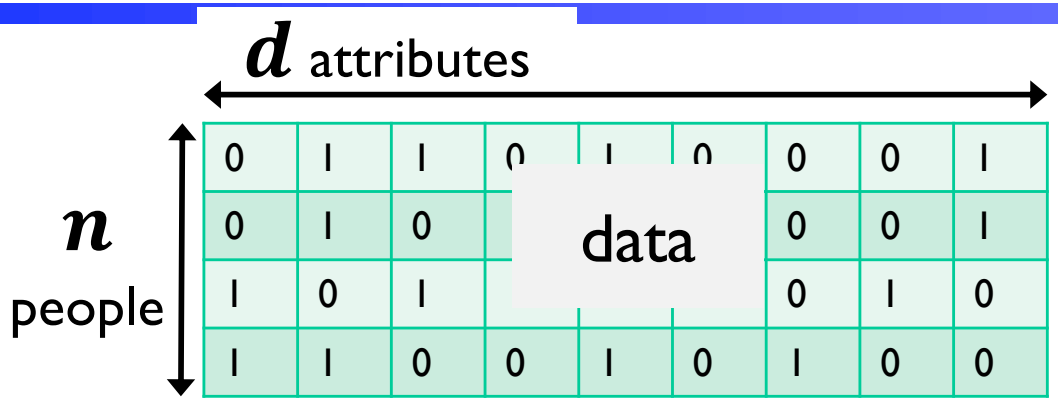
either flip or leave alone all bits

$\sum \tilde{z}_i$ allows estimating all frequencies with error $\approx \frac{1}{\varepsilon}\sqrt{\frac{\log d}{n}}$.

(2) Choose z_i pseudorandomly!

Nice connection to streaming heavy hitters.

Selection Lower Bounds



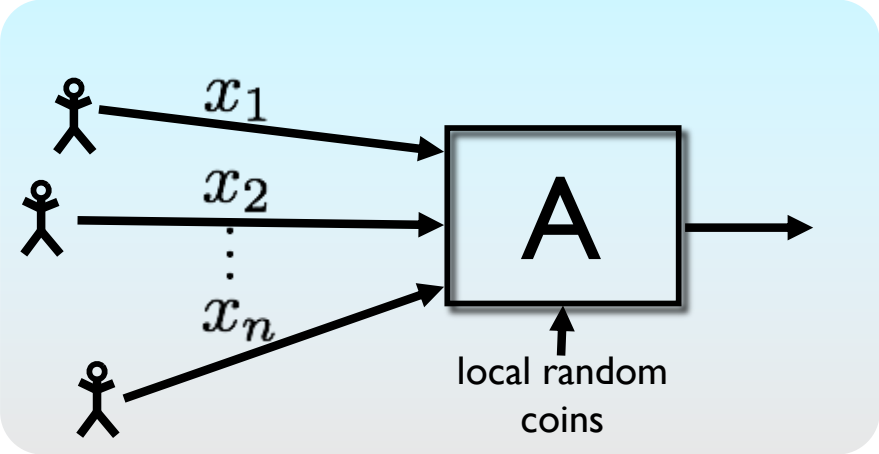
- Suppose each person has d binary attributes
- **Goal**: Find index j with highest count $(\pm\alpha)$

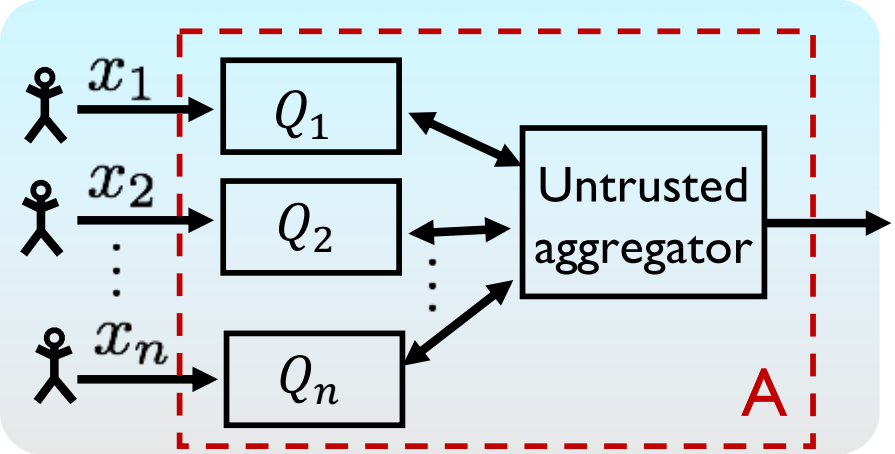
- **Central model**: $n = O(\log(d)/\varepsilon\alpha)$ suffices
  [McSherry Talwar '07]
- **Local model:** Any **noninteractive** local DP protocol with nontrivial error requires

$$n = \Omega(d \log(d) / \varepsilon^2 \alpha^2)$$

  - [DJW'13, Ullman '17]

Local Model for Privacy





What other models allow similarly distributed trust?

Distributed Models

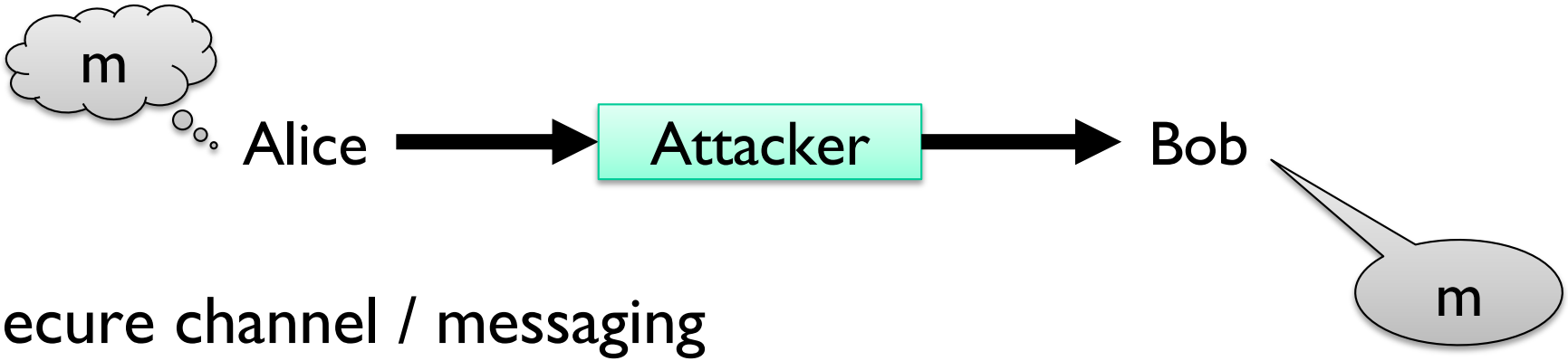
- Local Differential Privacy
  - Randomized Response Strikes Back
  - Limitations of the Model
- Cryptographic Tools
  - Encryption
  - Multiparty Computation
- What's next?
  - Efficient "federated" protocols?
  - Minimal crypto primitives?

Cryptography

- Powerful set of tools for controlling access to information and computation

- Two main aspects (for today)
  - Secure channels
  - Secure computation

Secure channels



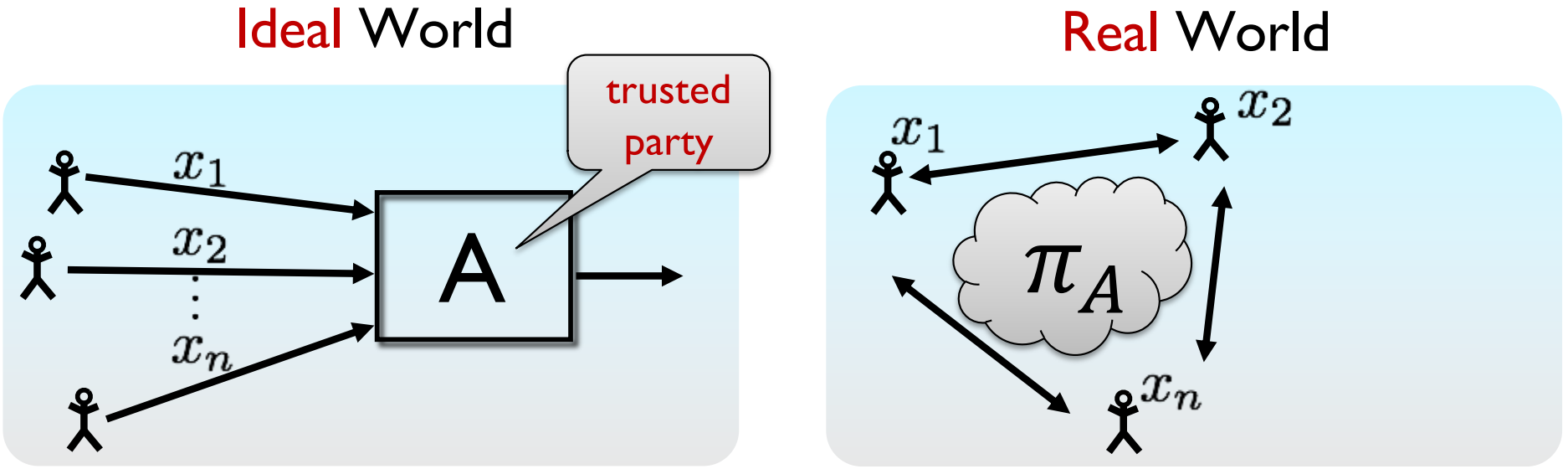
- Secure channel / messaging
  - Most widely used form of crypto
  - Think of Signal or WhatsApp
- Two main components
  - Encryption: ensure only a specific set of people can read a message
    - Only Bob can read Alice's email
  - Authentication: ensure that one of a specific set of people sent a message
    - Bob knows that Alice sent a message
- Security comes from secret, random keys
  - Requires infrastructure to generate and distribute keys

"Secure computation"

- Other cryptographic tools allow doing computations without directly seeing data, e.g.
  - Multiparty computation and secure function evaluation
  - Homomorphic encryption
  - Secure delegation
- Example applications:
  - BU wants to use Amazon servers to
    - Store its data
    - Process the data (e.g. generate monthly reports)

    … without letting Amazon see the data
  - Auction
    - Buyers submit bids
    - Everyone wants to learn who the winning bidder was
    - Auctioneer and winner should know the amount
  - Joint statistics
    - Boston-area businesses compute average gender salary gaps

Multiparty Computation [80's]



- Given an algorithm A with n inputs that we would like to run, an MPC protocol π_A for A allows n participants to
  - Execute A on their individual inputs $x_1, \ldots, x_n$
  - All receive the correct output a (given the inputs)
  - Reveal nothing except the information that is implied by a (and whatever subset of inputs the adversary knows)

… even when the adversary controls many of the participants

What secure computation does not provide

- Guarantees that participants only learn the output of the computation
  - e.g. auction winner, average wages

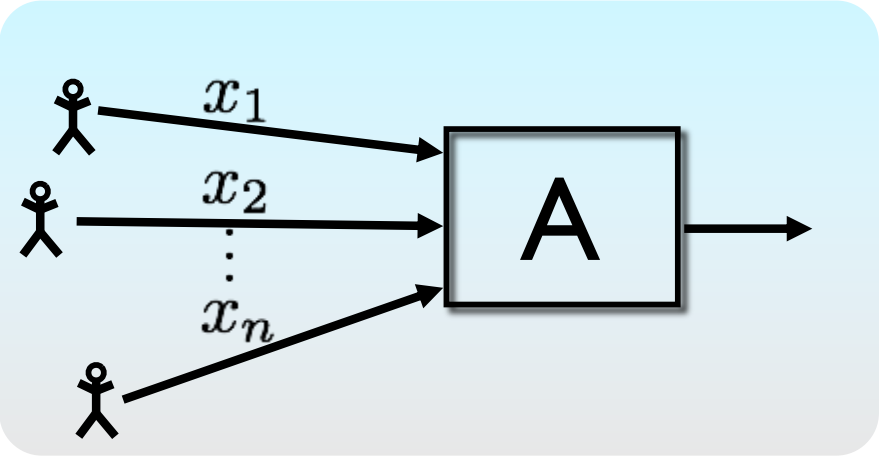
- No guarantees about what those outputs reveal
  - Auction winner learns upper bound on all other bids
  - Average salary before and after one resignation reveals that person's salary
  - ML models may leak training data

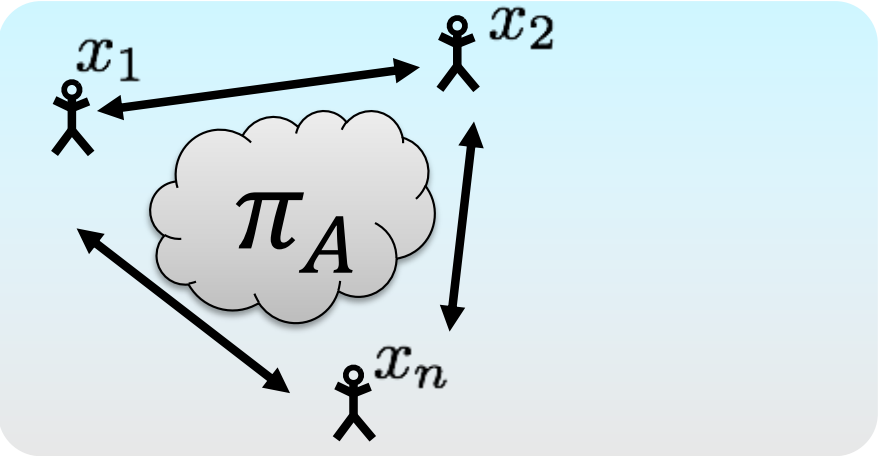
Privacy & Crypto

This course: privacy leakage of outputs

- Crypto: Works well when there are bright lines separating "inside" from "outside"
  - Psychiatrist and patient
  - Google and advertiser
- Data privacy: have to release some data at the expense of others
  - Different from "secure function evaluation"
  - SFE: how do we securely distribute a computation we've agreed on?
  - Data privacy: what computation should we perform?

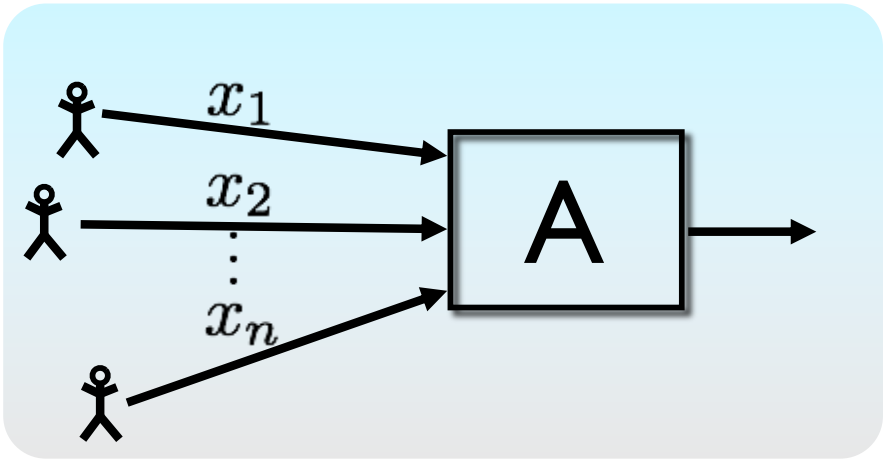
Two great tastes that go great together

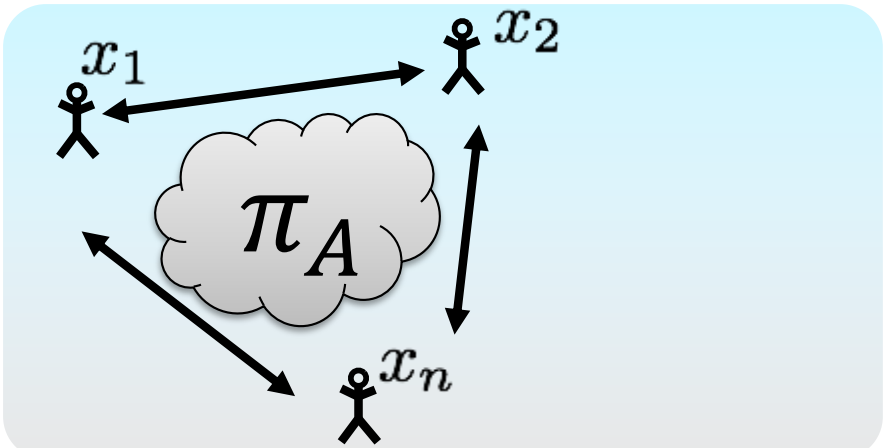




- How can we get accuracy without a trusted curator?
- Idea: Replace central algorithm A with multiparty computation (MPC) protocol for A (randomized), and either
  - Secure channels + honest majority
  - Computational assumptions + PKI
- Questions:
  - What definition does this achieve?
  - Are there special-purpose protocols that are more efficient than generic reductions?
  - What communication models make sense?
  - What primitives are needed?
    - Summation and "shuffling" are the most studied

Definitions





What definitions are achieved?

- Simulation of an (ε, δ)-DP protocol
- Computational DP [Mironov, Pandey, Reingold, Vadhan'08]

Not equivalent

Definition: A is (t, ε, δ)-computationally differentially private if,
for all neighbors $\boldsymbol{x}$, $\boldsymbol{x}'$,
for all distinguishers $\boldsymbol{T} \in time(t)$

$$\Pr_{\text{coins of } A}\left(\boldsymbol{T}\left(A(\boldsymbol{x})\right) = 1\right) \leq e^{\varepsilon} \cdot \Pr_{\text{coins of } A}\left(\boldsymbol{T}\left(A(\boldsymbol{x}')\right) = 1\right) + \delta$$

Distributed Models

- Local Differential Privacy
  - Randomized Response Strikes Back
  - Limitations of the Model
- Cryptographic Tools
  - Encryption
  - Multiparty Computation
- What's next?
  - Efficient "federated" protocols?
  - Minimal crypto primitives?