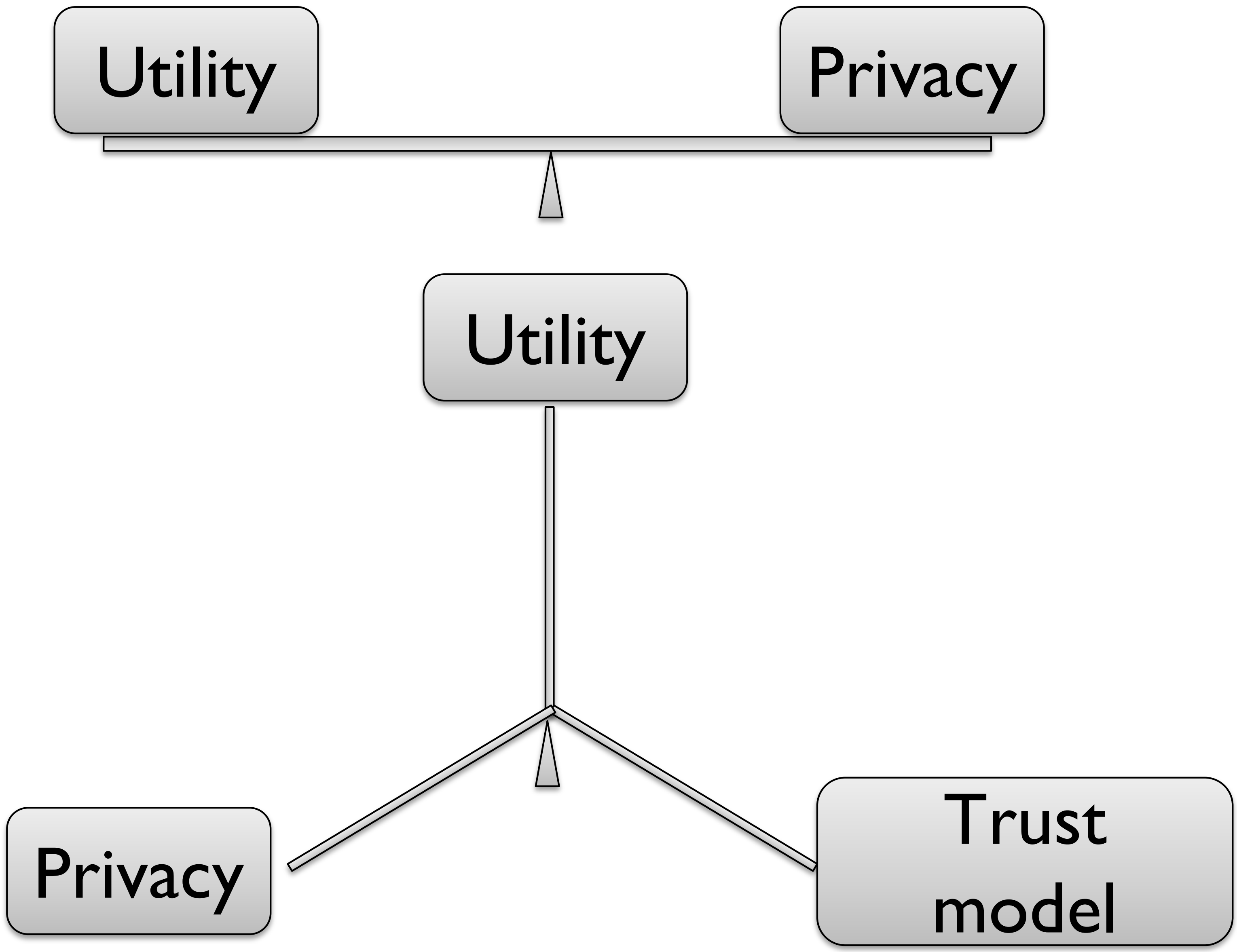
BU CS599
Spring 2023

Lecture 26: Distributed Models

Jonathan Ullman
NEU

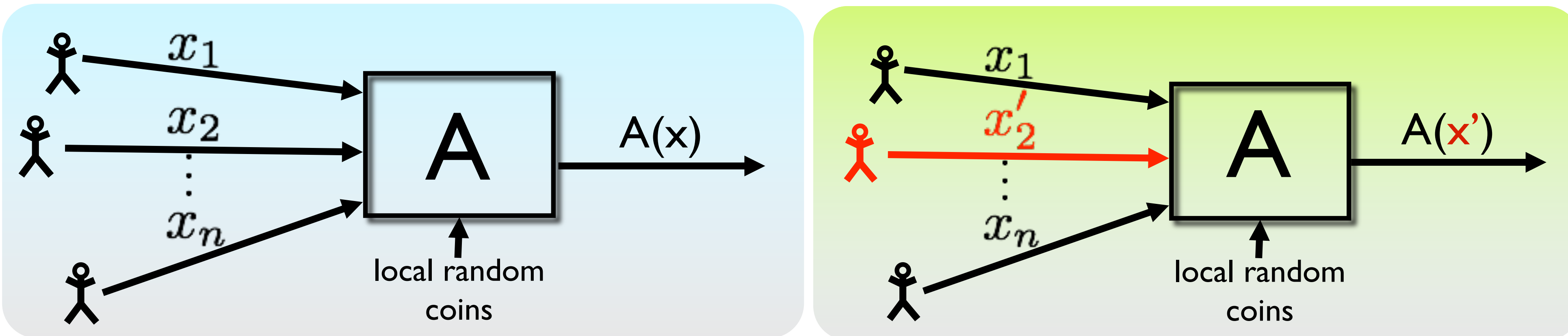
Adam Smith
BU


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



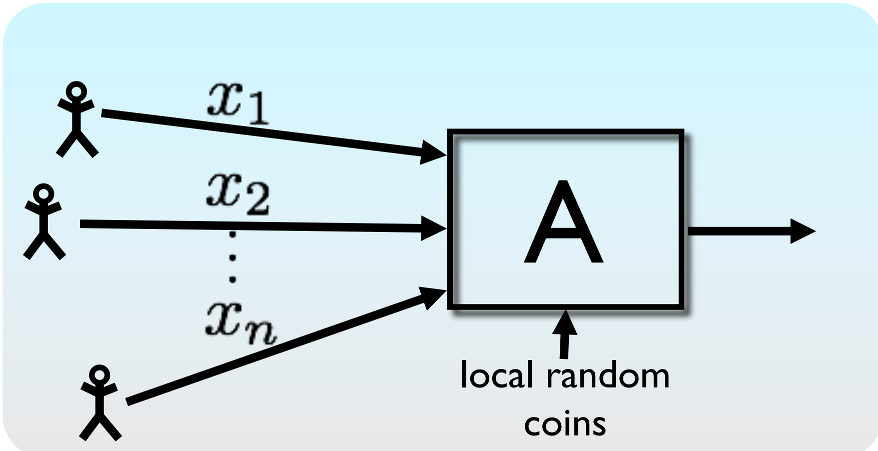
x' is a neighbor of *x*
if they differ in one data point

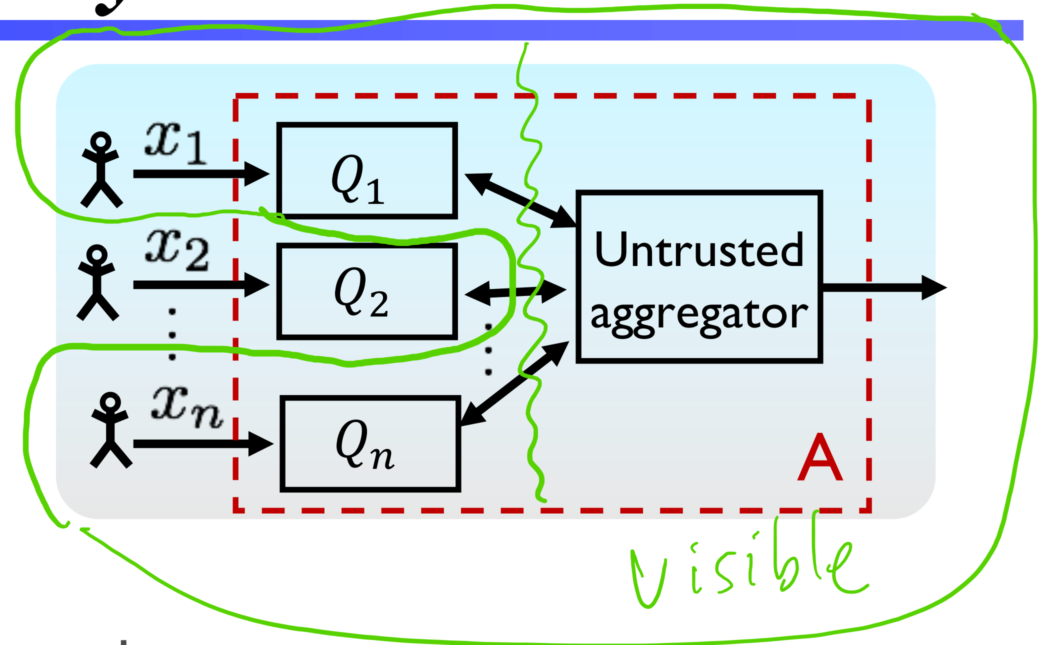
Neighboring databases induce **close** distributions on outputs

Definition: A is ϵ-differentially private if,
for all neighbors x, x',
for all sets of outputs T

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

Local Model for Privacy





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

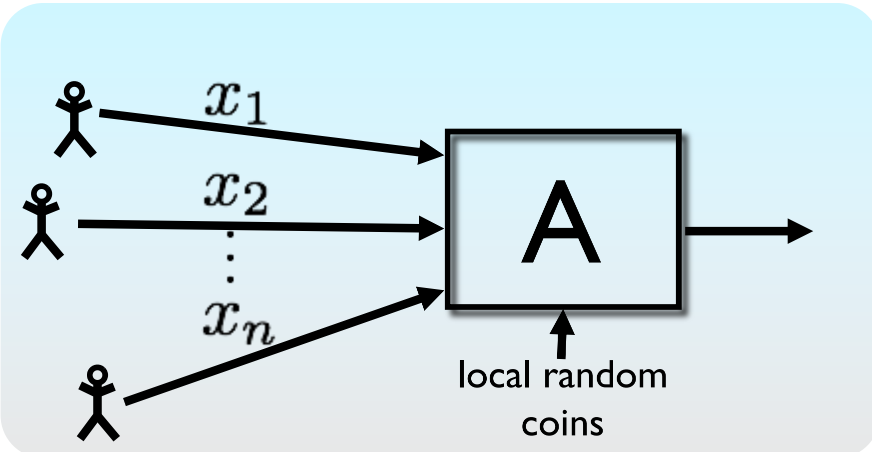
- Definition: A is ϵ-locally differentially private if for all i:
  - ➢ for all neighbors $\boldsymbol{x}$, $\boldsymbol{x}'$ that differ in position i
  - ➢ for all local coins $\boldsymbol{r}_{-i}$ of all other parties,
  - ➢ for all transcripts t:

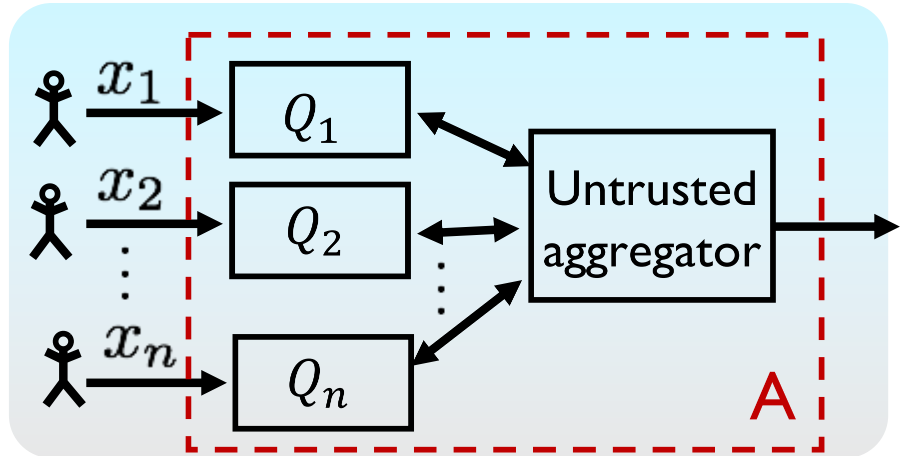
$$\Pr_{\text{coins } r_i}(A(\boldsymbol{x}, r_{-i}) = t) \le e^{\epsilon} \cdot \Pr_{\text{coins } r_i}(A(\boldsymbol{x}', r_{-i}) = t)$$

Reduces to: each Q_i ε-DP for $n = 1$.

$\delta = 0$ w.l.o.g.

Local Model for Privacy





- Pros
  - No trusted curator
  - No single point of failure
  - Highly distributed
  - Beautiful algorithms
- Cons

Selection: Central: $O\left(\frac{\log d}{\varepsilon n}\right)$

Local model: $\tilde{\Theta}\left(\frac{\sqrt{d}}{n}\right)$ (-ish)

  - Lower accuracy

randomized response (optimal)

Laplace.

    - Proportions: $\Theta\left(\frac{1}{\epsilon\sqrt{n}}\right)$ error [BMO'08,CSS'12] vs $O\left(\frac{1}{n\epsilon}\right)$ central
  - Correctness requires honesty (e.g. [CheuSU'21])

Reminder: Randomized response

∘∘ If person i always reports $Y_i = 1$ then they skew answer by $\approx \frac{1}{\varepsilon}$

- 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 $z \in \{0,1\}$:

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

add to 1

Idea: $I(X_i ; Q(X_i)) \lesssim \varepsilon^2$

$I(X_1 \cdots X_n ; Q(X_1), \dots, Q(X_n)) \lesssim \varepsilon^2 n$

Can set things up so that for accuracy α, $I(\dots) \lesssim \alpha^2 \varepsilon^2 n$.

- How can we estimate a proportion?
  - ➢ $A(x_1, \dots, x_n)$:
    - For each i, let $Y_i = R(\varphi(x_i))$
    - Return $A = \sum_i (aY_i - b)$ $\quad a \approx \frac{1}{\varepsilon}$
  - ➢ What values for a, b make $\mathbb{E}(A) = \sum_i \varphi(x_i)$?
- **Proposition:** $\sqrt{\mathbb{E}\left(A - \sum_i \varphi(x_i)\right)^2} \leq \frac{e^{\epsilon/2}}{e^{\epsilon}-1}\sqrt{n}.$

$\approx \frac{\sqrt{n}}{\epsilon}$ when ϵ small

Case Study: Histograms/Heavy Hitters

- Inputs: $x_1, \dots, x_n \in [d]$
- Goal: Find $n_1, n_2, \dots, n_d \in \mathbb{N}$, where $n_j = \#\{i : x_i = j\}$
- How can use RR?

i) Suppose all $x_i = j^*$ . .
- In each position, all bits equal
- So $n \approx \frac{1}{\varepsilon'^2}$ records suffice to identify the bit.

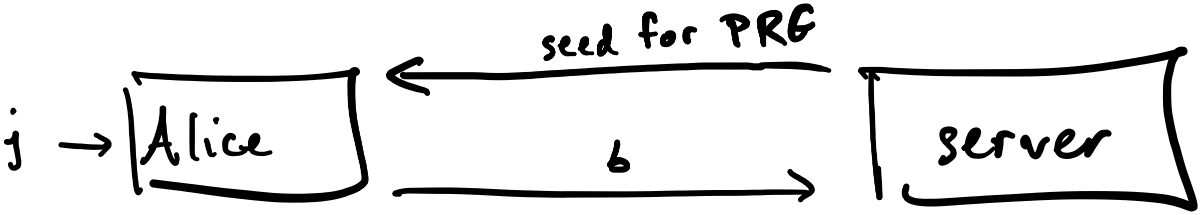
$\approx \frac{\log d}{\varepsilon^2}$

1. Randomized the input directly:
   a) Write each x_i as string in $\{0,1\}^{\log d}$
   use composition b) Apply $RR_{\epsilon'}$ to each bit (for $\epsilon' \approx \epsilon/\cancel{d}$ $\sqrt{\log_2 d}$)

(ii) Terrible with more general inputs

2. How about randomize "one-hot encoding" of x

- Map $[d] \to \{0,1\}^d$ where $x \mapsto (0, 0, \dots, 0, \underset{x}{1}, 0, \dots, 0)$
- Apply $RR_{\varepsilon/2}$ to each position
- expected error $\approx \frac{1}{\varepsilon}\sqrt{n \log d}$
- High communication



$$Y_j = \begin{cases} z & \text{if } b=0 \\ 1-z & \text{if } b=1. \end{cases}$$

$z = PRG(\text{seed})$
length d.

$b = RR_\varepsilon(1 - z_j)$

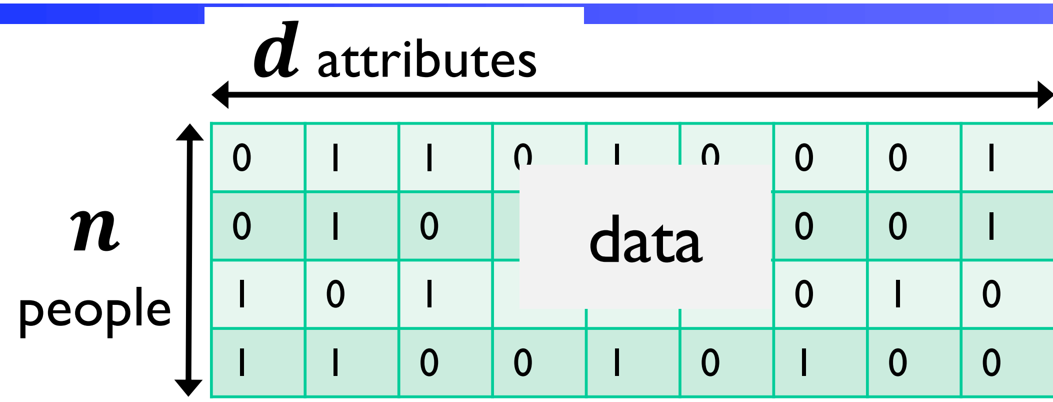
1) Privacy : unchanged. (ε - local DP)

2) Accuracy : suffices for entries of z to be pairwise indep and uniform.

To get efficiency : apply protocol repeatedly to longer and longer prefixes of binary encodings.

(run time from d down to $\approx \log_2 d$)

Selection Lower Bounds



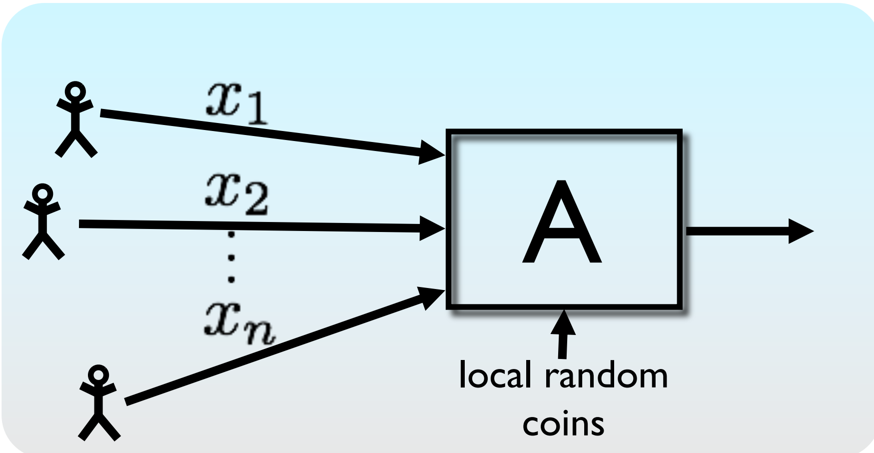
- Suppose each person has d binary attributes
- **Goal**: Find index j with highest count $(\pm\alpha)$

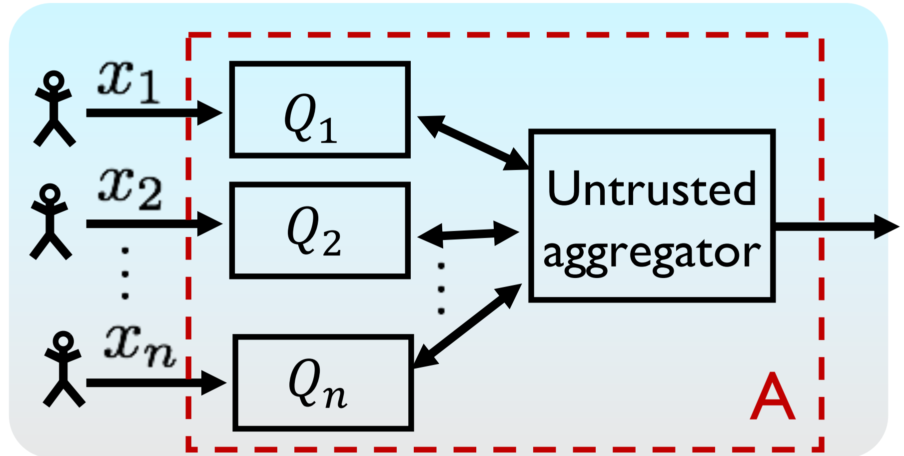
- **Central model**: $n = O(\log(d)/\epsilon\alpha)$ suffices
  [McSherry Talwar '07]
- **Local model:** Any **noninteractive** local DP protocol with nontrivial error requires

$$n = \Omega(d\log(d)\,/\epsilon^2)$$

  - [DJW'13, Ullman '17]

Local Model for Privacy





What other models allow similarly distributed trust?

Distributed Models

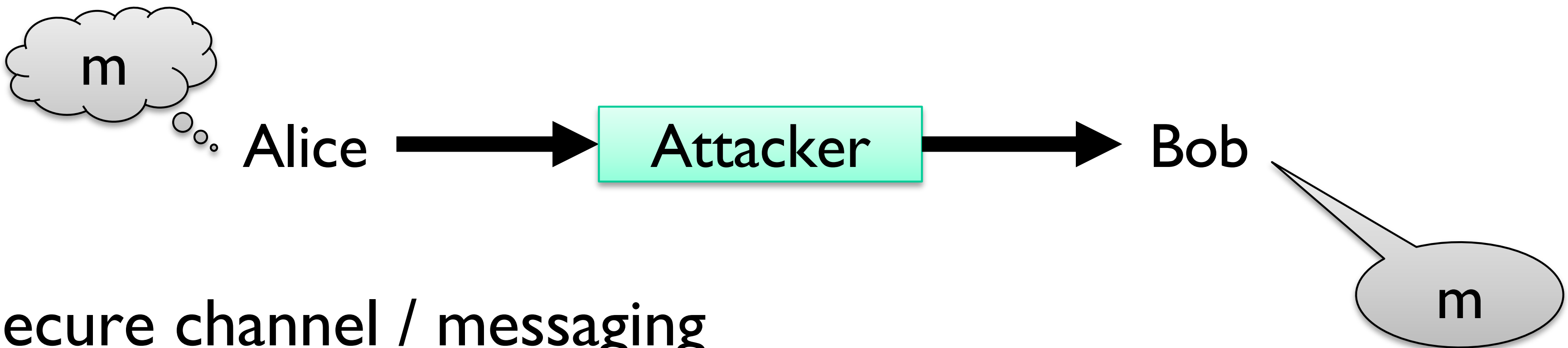
- Local Differential Privacy
  - Randomized Response Strikes Back
  - Limitations of the Model
- Cryptographic Tools
  - Encryption
  - Multiparty Computation
- What's next?
  - Efficient "federated" protocols?
  - Minimal crypto primitives?

Cryptography

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

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

Secure channels



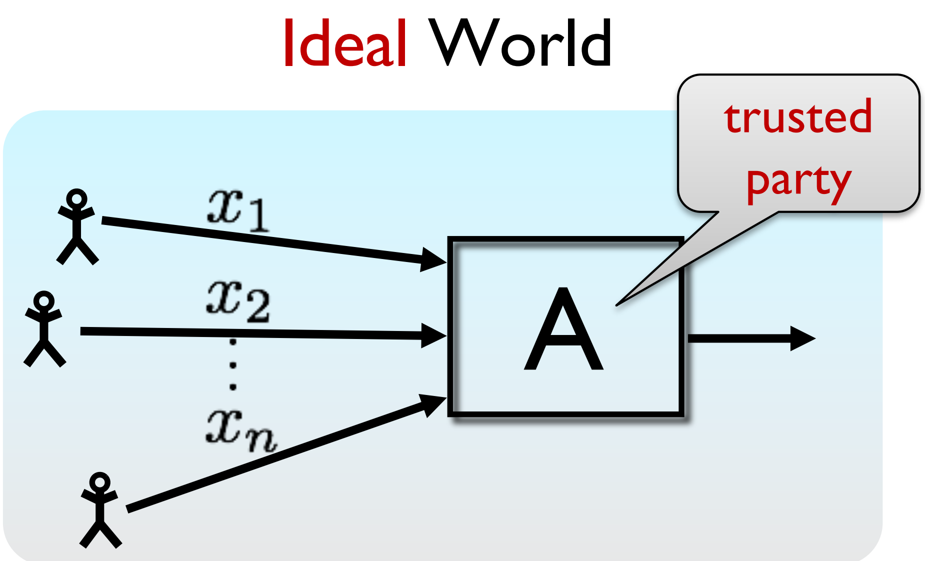
- Secure channel / messaging
  - Most widely used form of crypto
  - Think of Signal or WhatsApp
- Two main components
  - Encryption: ensure only a specific set of people can read a message
    - Only Bob can reads Alice's email
  - Authentication: ensure that one of a specific set of people sent a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

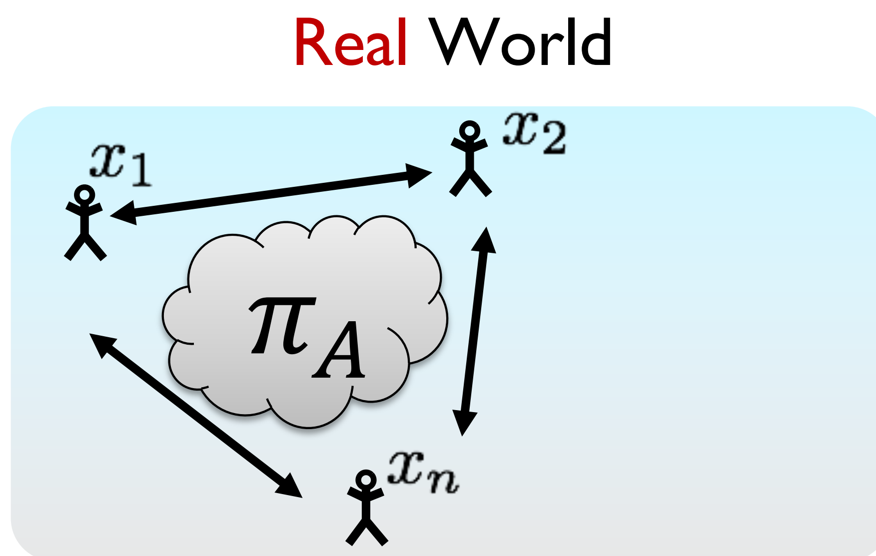
“Secure computation”

- 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 $x_1, \dots, x_n$
  - 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

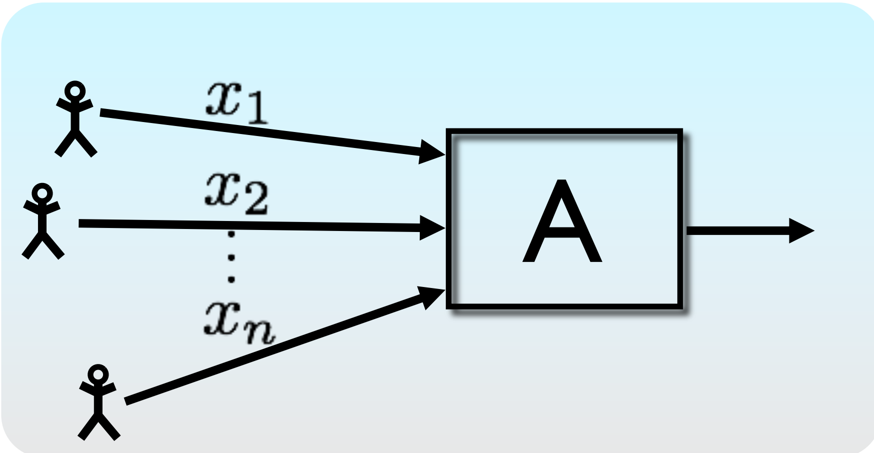
Privacy & Crypto

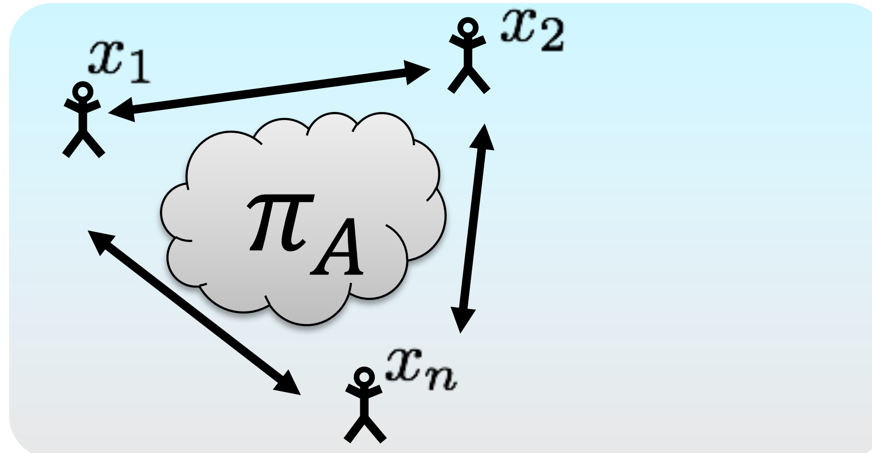
This course: privacy leakage of outputs

- 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?

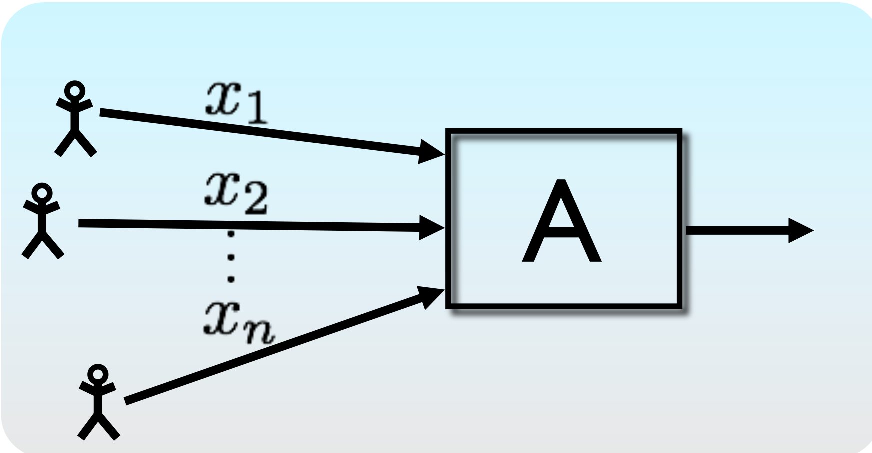
Two great tastes that go great together

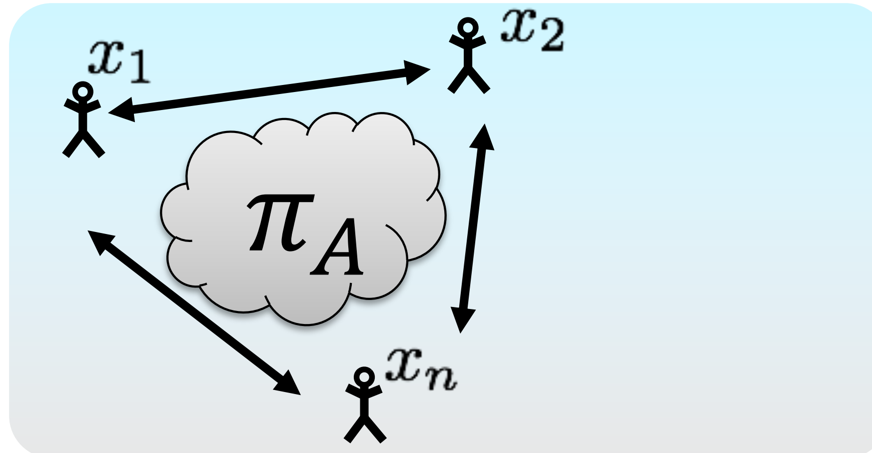




- How can we get accuracy without a trusted curator?
- Idea: Replace central algorithm A with multiparty computation (MPC) protocol for A (randomized), and either
  - Secure channels + honest majority
  - Computational assumptions + PKI
- Questions:
  - What definition does this achieve?
  - Are there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

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 x, x',
for all distinguishers $T \in time(t)$

$$\Pr_{\text{coins of } A}\big(T\big(A(x)\big) = 1\big) \leq e^{\epsilon} \cdot \Pr_{\text{coins of } A}\big(T\big(A(x')\big) = 1\big) + \delta$$

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?