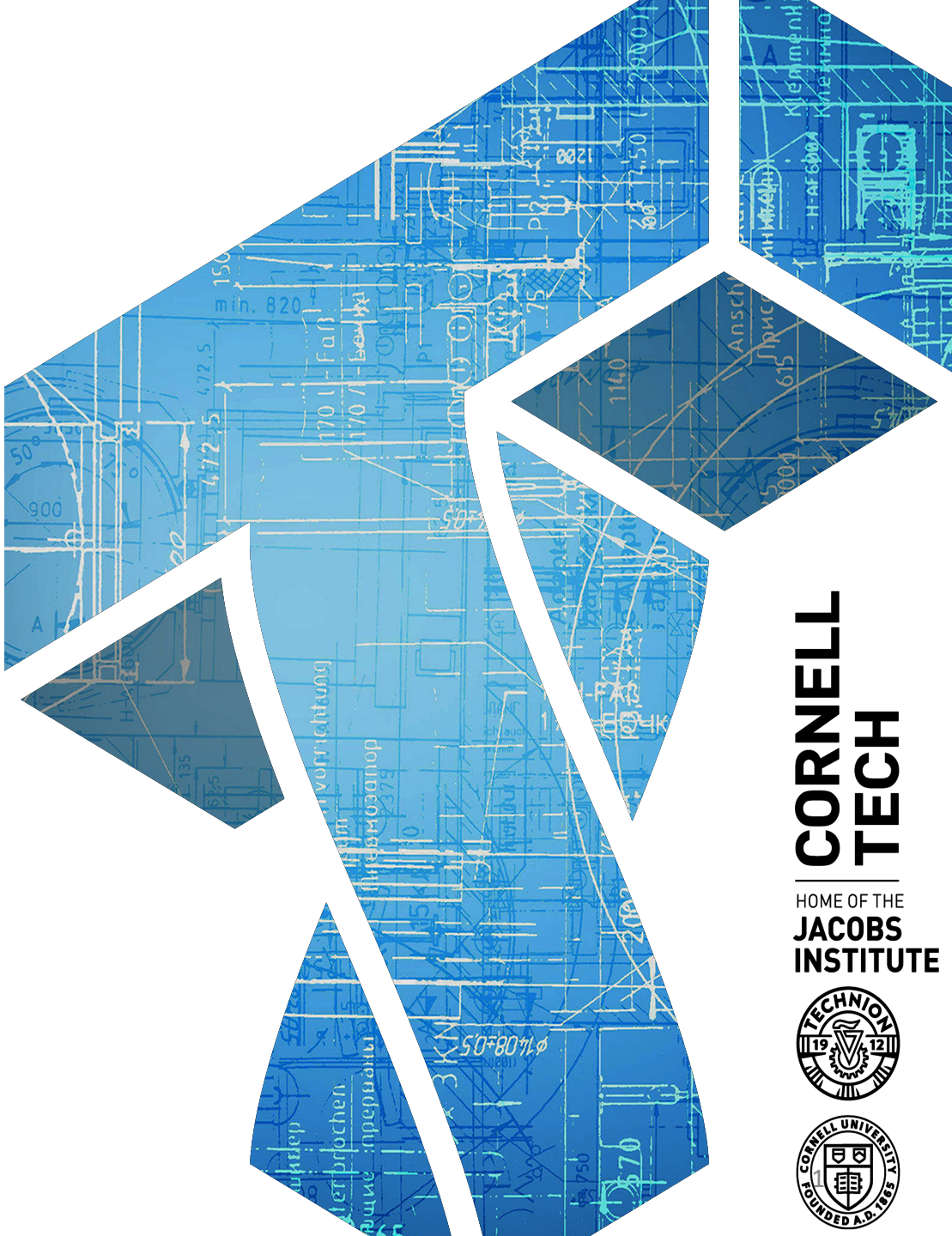
# CS 6431: Differential privacy

Instructor: Tom Ristenpart

https://github.com/tomrist/cs6431-fall2021



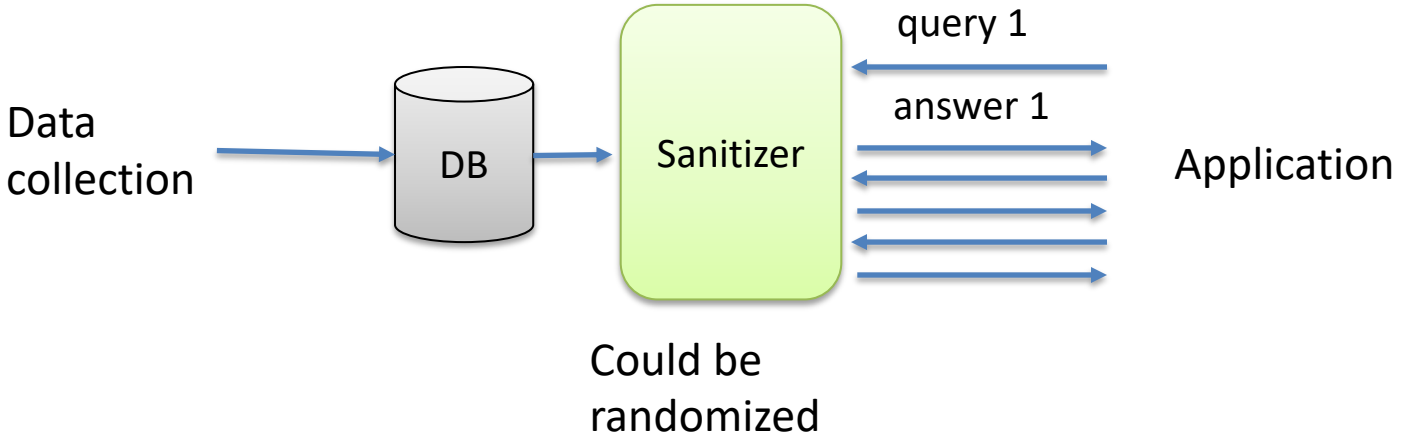
CORNELL TECH
HOME OF THE JACOBS INSTITUTE

# Privacy and re-identification

- Sanitization removes PII and other explicit identifiers
- Sweeney's results about pseudo-identifiers
- K-anonymity and variants seek to prevent re-identification attacks
  - Various deficiencies (settings in which they do not work)

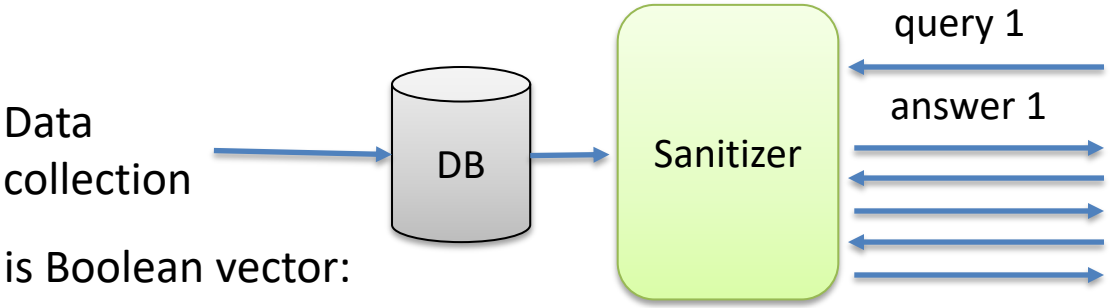
- Today: the differential privacy approach

# The sanitization model




- ***Input perturbation***: add noise to DB and then release
- ***Summary statistics***: just release aggregate stats (means, variances, marginal totals, regression coefficients)
- ***Output perturbation***: add noise to statistics
- ***Interactive versions of above***: can decide how/if to answer queries
  - ***Query auditing***: track which queries have happened and disallow risky ones

# [Dinur-Nissim 02] result




Database is Boolean vector:

$$d = (d_1, \ldots, d_n)$$

Could be randomized

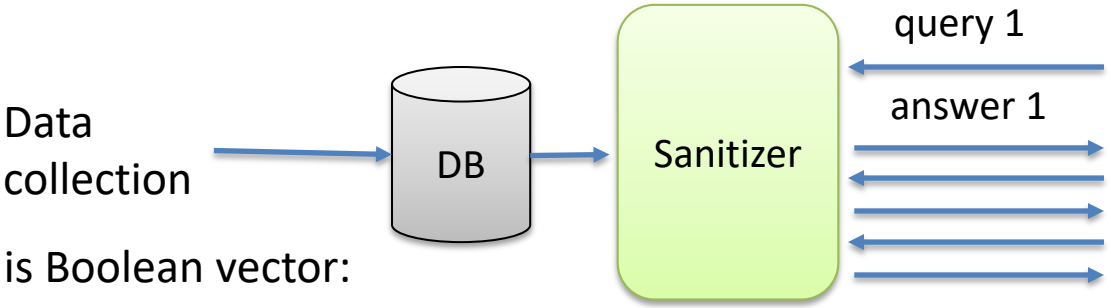
Queries are subset sums:

$$a_q = \sum_{i \in q} d_i$$

Database non-privacy definition (informal):
Good probability that adversary outputs d' with $\mathrm{dist}(d', d) < \epsilon n$
using some reasonable amount of queries

Assume queries answered directly. How do we recover database?

# [Dinur-Nissim 02] result




Database is Boolean vector:

$$d = (d_1, \ldots, d_n)$$

Could be randomized

Queries are subset sums:

$$a_q = \left( \sum_{i \in q} d_i \right) + \mathcal{E}$$

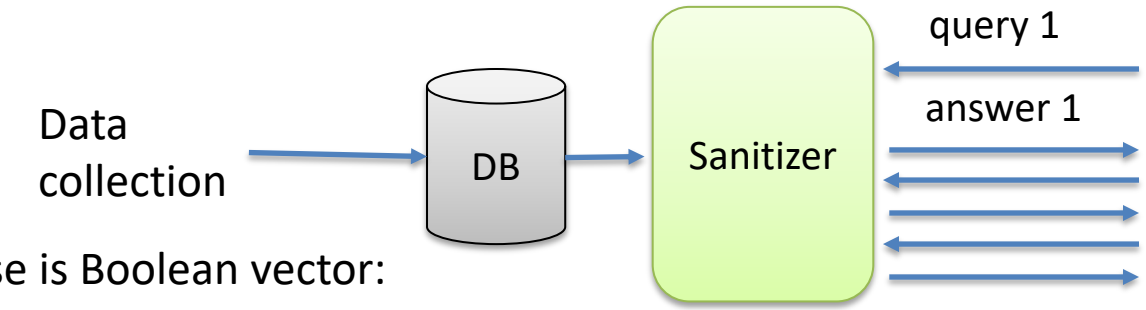
Noise perturbation, chosen i.i.d. for each query

Database non-privacy definition (informal):
Good probability that adversary outputs d' with $\mathrm{dist}(d', d) < \epsilon n$
using some reasonable amount of queries

How do we break this? "Averaging" attack

Can attempt to fix by disallowing repeat queries

# [Dinur-Nissim 02] result




Database is Boolean vector:

$$d = (d_1, \ldots, d_n)$$

Could be randomized

Queries are subset sums:

$$a_q = \left( \sum_{i \in q} d_i \right) + \mathcal{E}$$

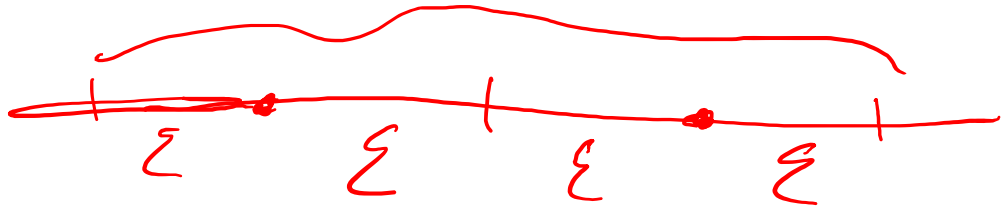
Noise perturbation, chosen i.i.d. for each query

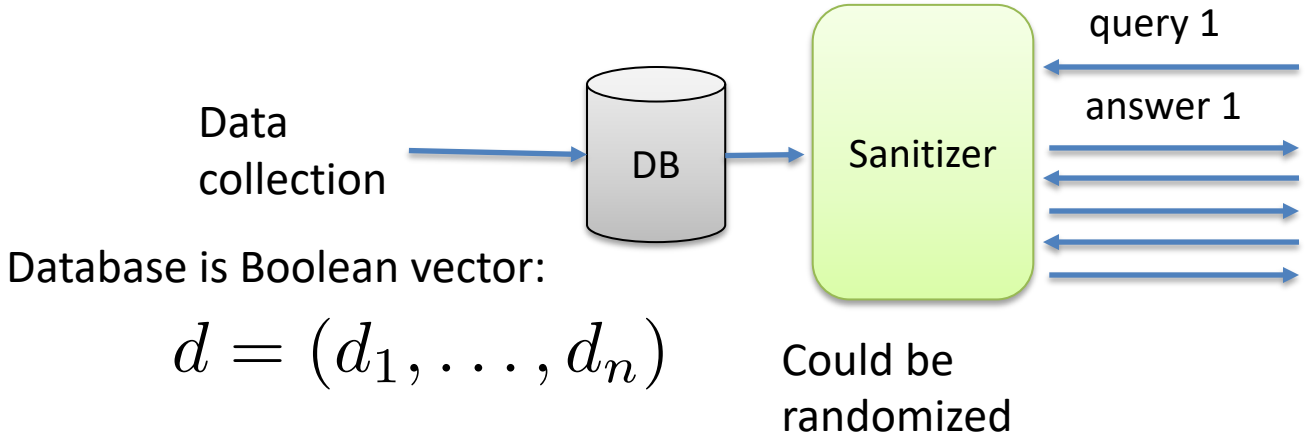
Database non-privacy definition (informal):
Good probability that adversary outputs d' with $\mathrm{dist}(d', d) < \epsilon n$ $\epsilon = \max \mathcal{E}$
using some reasonable amount of queries

Result one: given $2^n$ queries, can recover database with $\mathrm{dist}(d', d) \leq 4\epsilon$

Result two: given n $\log^2$ n random queries, can recover database when $\epsilon = o(\sqrt{n})$

# [Dinur-Nissim 02] result




Queries are subset sums:

$$a_q = \left( \sum_{i \in q} d_i \right) + \mathcal{E}$$

Noise perturbation, chosen i.i.d. for each query

$$\epsilon = \max \mathcal{E}$$

Make all $2^n$ queries. Query q has noised answer $\tilde{a}_q$

Find candidate database d' s.t. for all queries q: $\left| \left( \sum_{i \in q} d'_i \right) - \tilde{a}_q \right| \leq \epsilon$

Short argument shows that d' can't have dist(d',d) > 4$\epsilon$

# Dalenius' Desideratum

Anything that can be learned about a respondent from a statistical database should be learnable without access to the database

Very similar to ***semantic security*** definition from cryptography: Whatever can be efficiently computed about a plaintext, given a ciphertext, should be computable without the ciphertext

What problems with this definition does paper discuss?

# A viewpoint shift

- Instead, focus on ***membership privacy***
  - Privacy is about whether or not an individual is in a database

- What are pros of this?

- What are cons of this?

# Differential privacy

[Dwork, McSherry, Nissim, Smith '06]

Build randomized mechanisms for rendering database (or queries against a database) s.t. adversary can't with high confidence know if any particular individual in data set

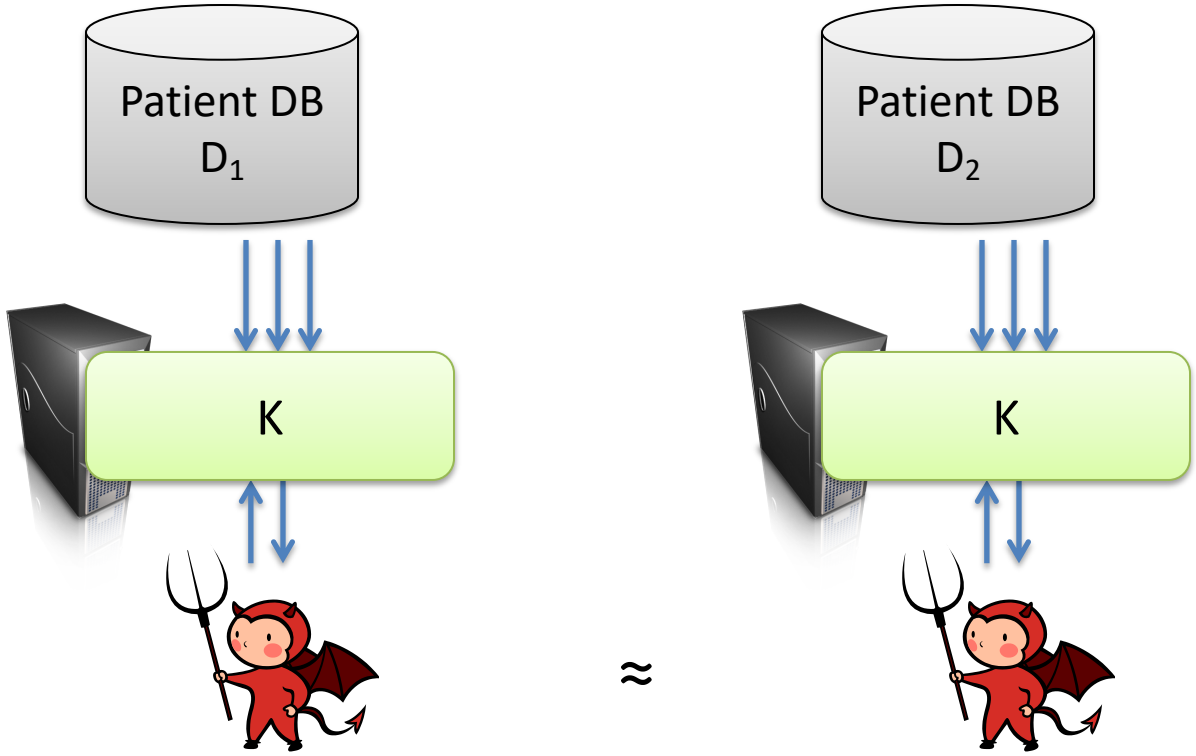
K is randomized algorithm. K is ***ε-differentially private*** if:
for all datasets $D_1$ , $D_2$ differing in at most one row and all subsets $S$ of K's range:

$$\Pr[\mathrm{K}(D_1) \in S] \leq e^{\epsilon} \cdot \Pr[\mathrm{K}(D_2) \in S]$$


K *necessarily* adds noise, forcing a utility vs. privacy trade-off

# Differential privacy

[Dwork, McSherry, Nissim, Smith ‘06]



# DP definition properties

- Worst-case over databases (within distance 1)
  - Adversary knows databases; models worst-case adversarial auxiliary knowledge about a database
- Postprocessing is DP
  - Any computation on DP input provides DP output
- Composes
  - Two releases with $\epsilon$-DP is $2\epsilon$-DP
  - Same view gives group privacy
- "Protection against arbitrary risks, moving beyond protection against re-identification." [Dwork, Roth DP book]
  - They mean it isn't tailored to particular attacks

# Randomized response example

Randomized response example
Database is vector of bits $b_1$ , … , $b_n$ from n respondents

Mechanism K
For each entry:
1. Flip bit d at random
2. If d = 1, return random value
3. If d = 0, return true bit value

This yields an $\epsilon$-DP mechanism for $\epsilon$ = ln 3
Consider any entry $b_i$ and let true value be b

$$\Pr[b_i = b] = 3/4$$
$$\Pr[b_i = 1-b] = 1/4$$

Ratio is 3

This kind of mechanism sometimes called “local differential privacy”,
because the noise can be added locally before collection of data

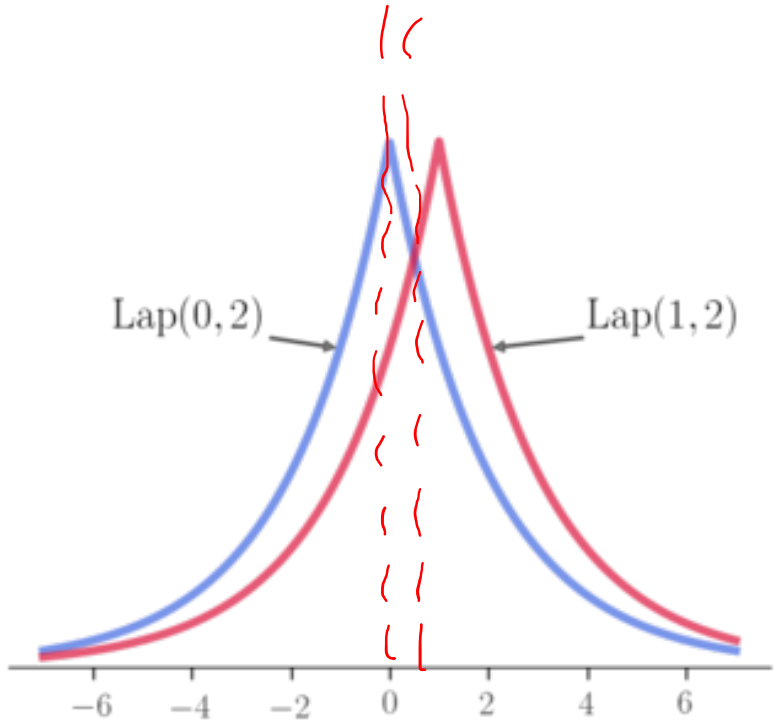
# Differential privacy

[Dwork, McSherry, Nissim, Smith '06]

Say we only want to render private some ***statistic*** of DB
Let f be function computing the statistic (e.g., median)

$$K(D,f,\varepsilon) = f(D) + Lap(0,\Delta f/\varepsilon)$$

where $\Delta f = \max|f(D_1) - f(D_2)|$
(over $D_1$ , $D_2$ differing in at most one row) is sensitivity of f
and $Lap(0,\Delta f/\varepsilon)$ is random according to Laplacian distribution




$$Lap(x|\mu,b) = \frac{1}{2b} e^{\left(\frac{|x-\mu|}{b}\right)}$$

***Intuition:*** add sufficient random noise centered on actual value to ensure uncertainty about which database used

# Rappor system in Chrome browser

Google wants to collect information on websites visited by users of Chrome

- [Elringsson et al. 2014] gives DP system for sending reports
- Builds on ***randomized response***. Flip a coin:
  - Heads: report "Visited example.com"  no matter what
  - Tails: report whether this user visited example.com
- Uses variant with deniability for both Yes/No answers, long-term private response, short-term private response, …
- Rappor uses $\varepsilon = \ln 3$

- Various limitations, see paper

# DP at Apple

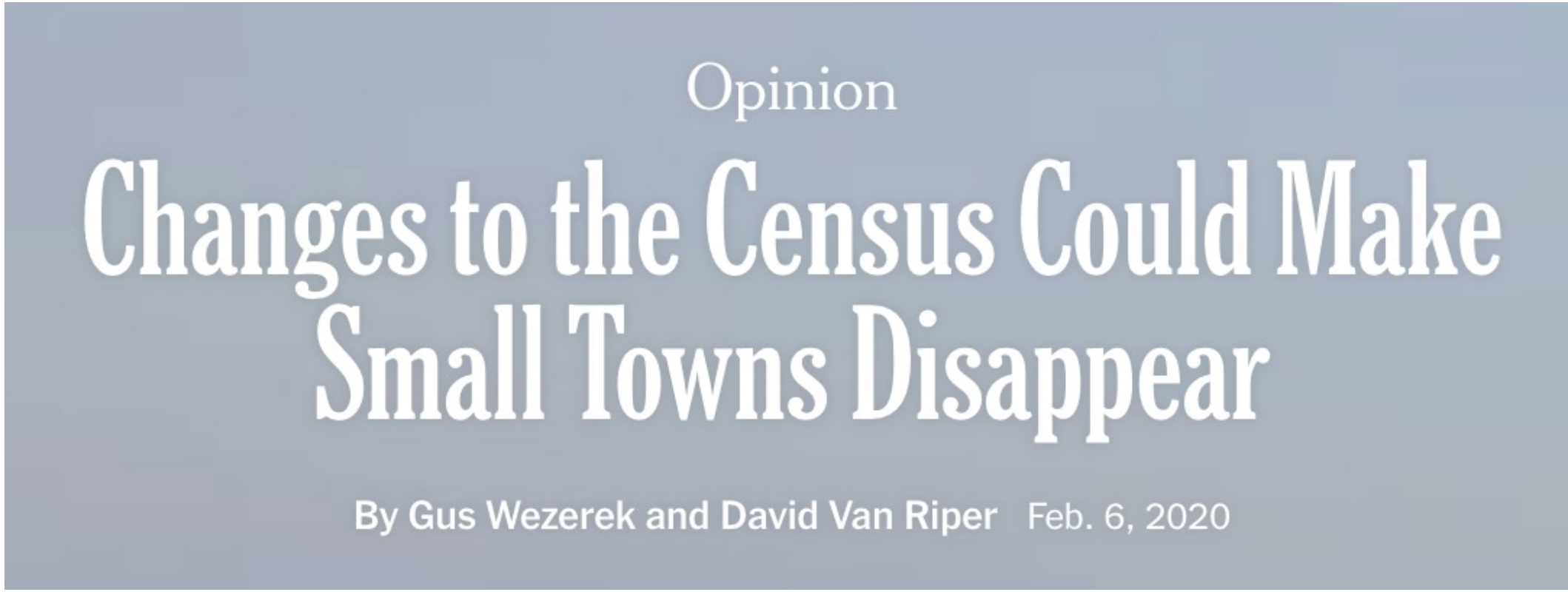
Apple deployed DP mechanism for certain user data items collected by iOS

- Proprietary implementation
- [Tang et al. 2017] reverse engineered implementation:
  - Privacy budget epsilon ε not sufficient
  - Re-upped privacy budget each day
- Unclear how much privacy protection really offered

# DP for the 2020 US census data releases

- Prior releases used swapping between census blocks to help prevent re-identification risk
  - Re-identification experiment with 2010 data re-identified 17% of the 309 million records
- Decision to move to using DP for 2020 census public datasets
- TopDown algorithm
  - [Abowd et al. 2019] https://systems.cs.columbia.edu/private-systems-class/papers/Abowd2019Census.pdf
- Many concerns over implications for data use

# DP for the 2020 US census data releases




https://www.nytimes.com/interactive/2020/02/06/opinion/census-algorithm-privacy.html

# Data privacy is hard

- Balancing utility versus privacy risks
- Differential privacy has come to dominate viewpoint
  - Membership privacy
- Concern companies use it as smokescreen for invasive data collection, without addressing real societal problems
- Alternative viewpoints, such as contextual integrity, speak more to privacy as data flows