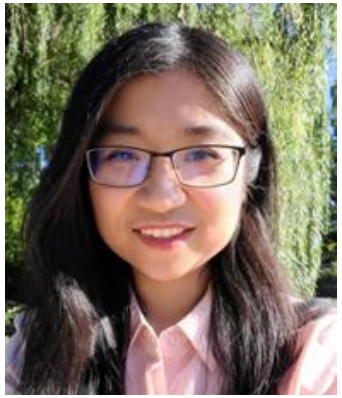
Edge Differentially Private Triangle Counting in the Local Model

Sofya Raskhodnikova
Boston University

Joint work with



Talya Eden
Bar-Ilan University

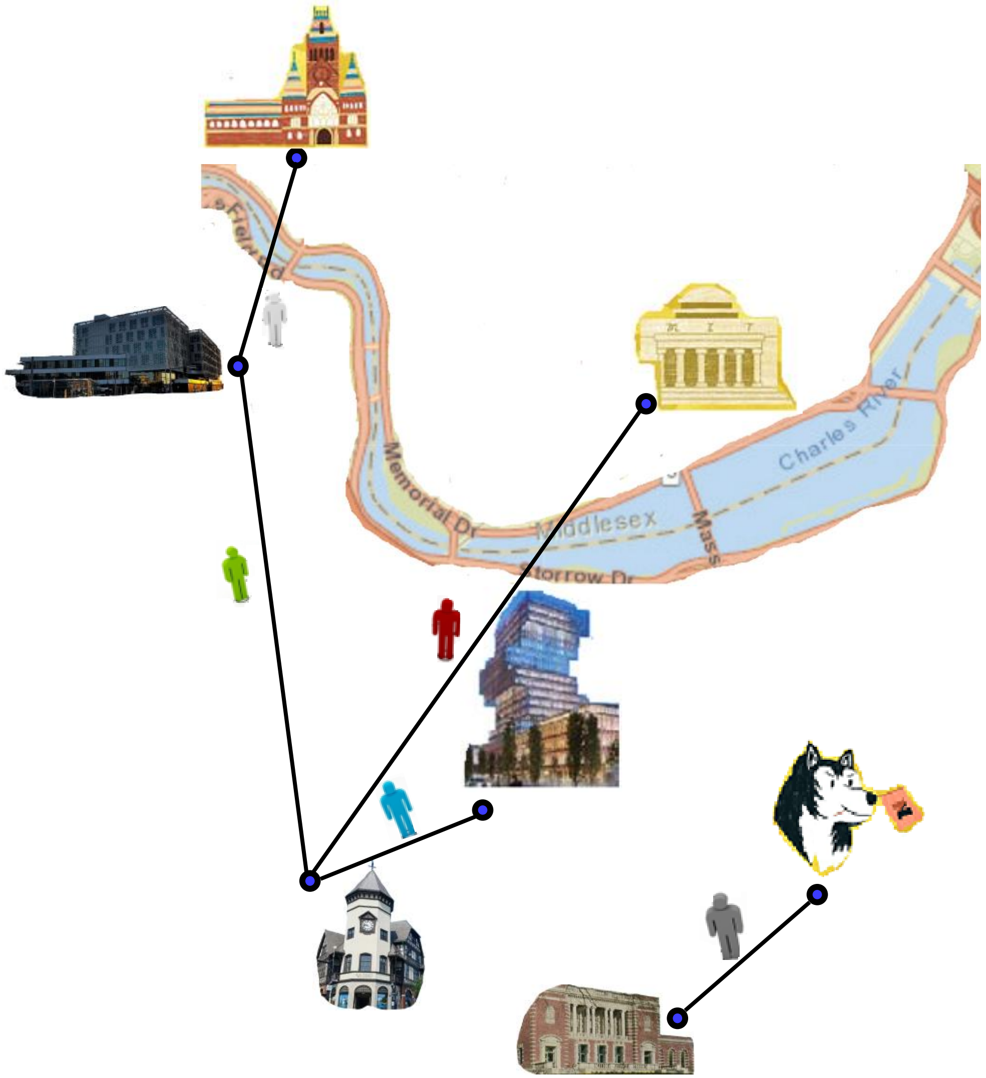
Quanquan Liu
Northwestern University

Adam Smith
Boston University

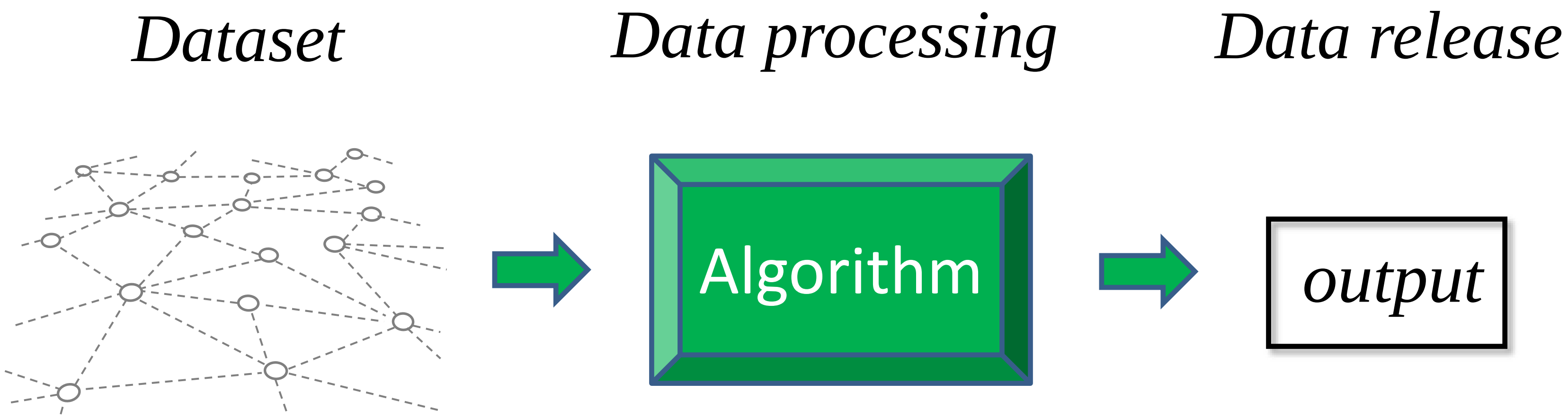
BOSTON UNIVERSITY

Publishing information about graphs

Many types of sensitive data can be represented as graphs



Differential privacy



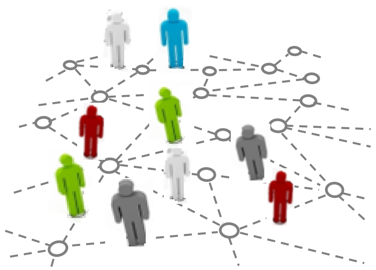
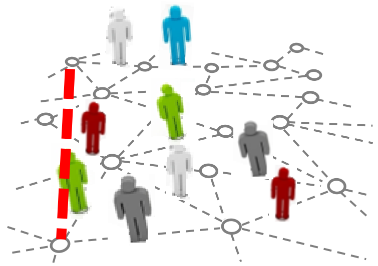
Differential privacy [Dwork McSherry Nissim Smith 06]

Intuition: Two datasets are ***neighbors*** if they differ in one individual’s data.

An algorithm is **differentially private** if its output is roughly the same for all pairs of ***neighbors.***

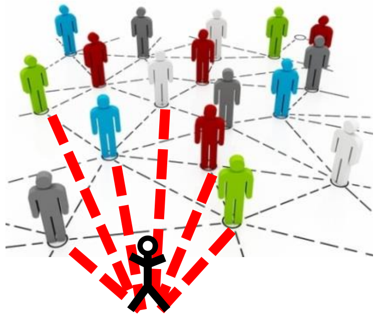
Two variants of differential privacy for graphs

- **Edge** differential privacy

***G*:**  ***G′*:** 

Two graphs are **neighbors** if they differ in ***one edge***.

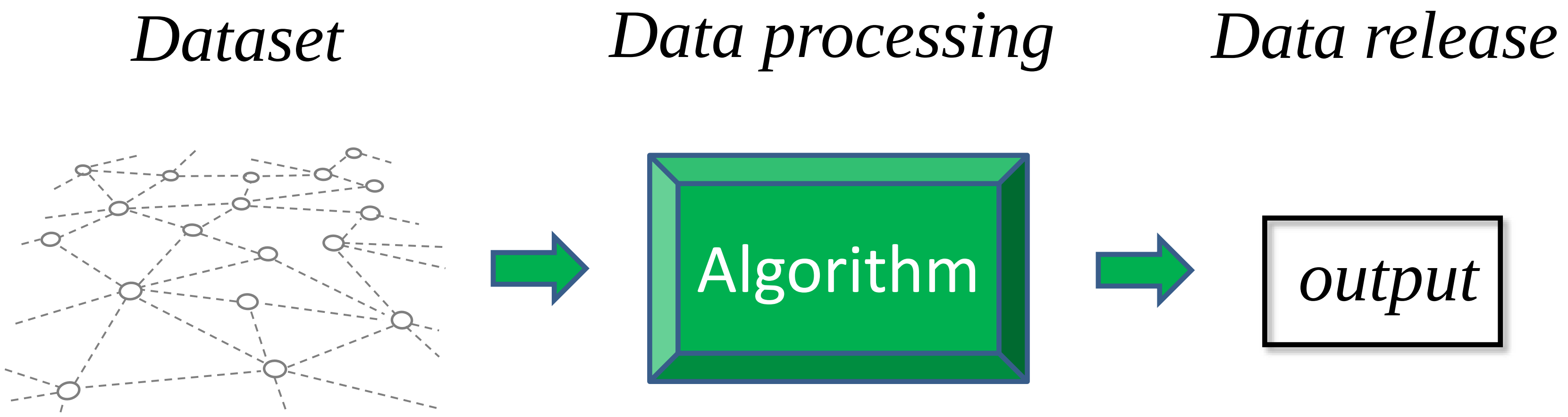
- **Node** differential privacy

***G*:**  ***G′*:** 

Two graphs are **neighbors** if one can be obtained from the other by deleting ***a node and its adjacent edges***.

image source http://www.queticointernetmarketing.com/new-amazing-facebook-photo-mapper/

Differential privacy (for graph data)



Differential privacy [Dwork McSherry Nissim Smith 06, Nissim Raskhodnikova Smith 07]

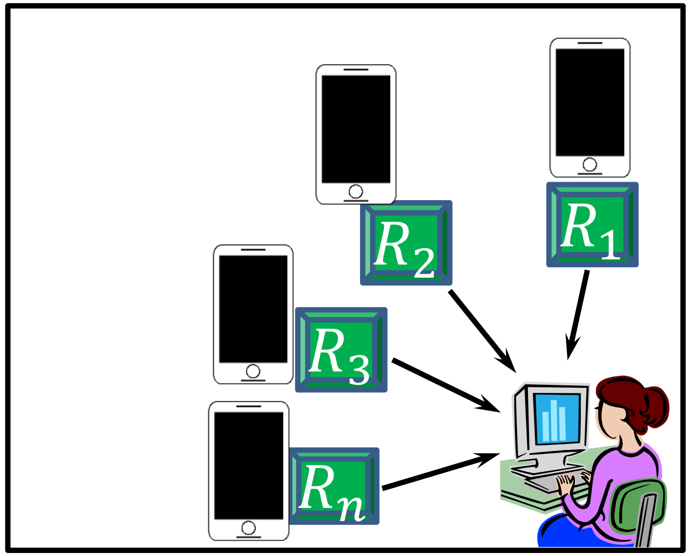
An algorithm A is **(ϵ, δ)-differentially private** if
for all pairs of ***neighbors*** $\boldsymbol{G}, \boldsymbol{G'}$ and all possible sets of outputs S:

$$\mathbf{Pr}[\boldsymbol{A}(\boldsymbol{G}) \in \boldsymbol{S}] \leq \boldsymbol{e}^{\epsilon}\, \mathbf{Pr}[\boldsymbol{A}(\boldsymbol{G'}) \in \boldsymbol{S}] + \boldsymbol{\delta}$$

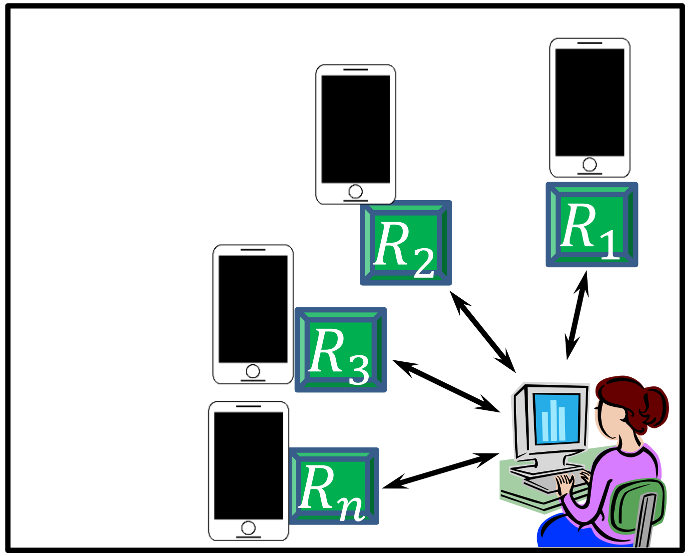
Local Privacy Models

[Efvimievski Gehrke Srikant 03]
[Kasiviswanathan Lee Nissim Raskhodnikova Smith 11]

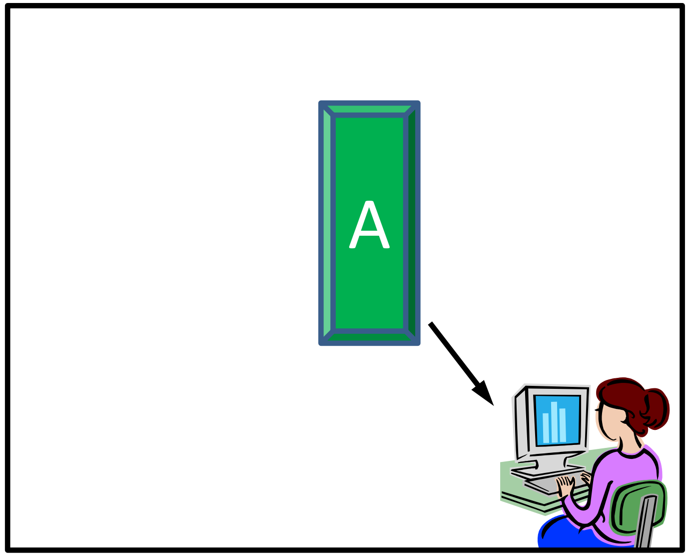
Local Noninteractive



Local (Interactive)



Centralized

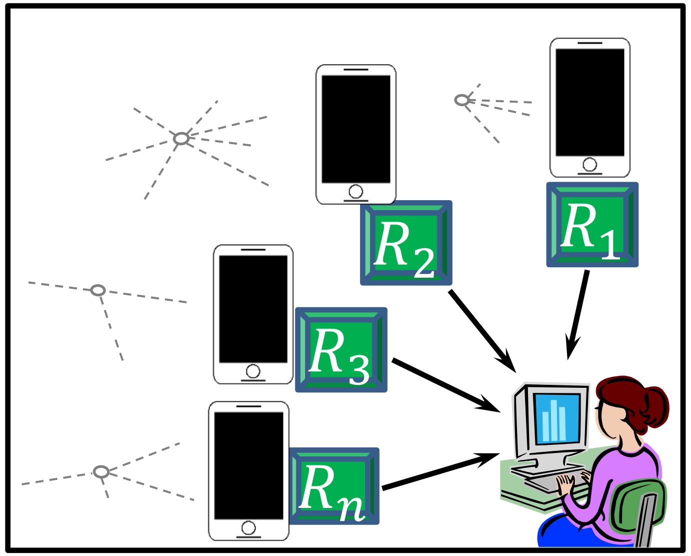


- Advantages of the local model:
  - private data never leaves local devices
  - no single point of failure
  - highly distributed

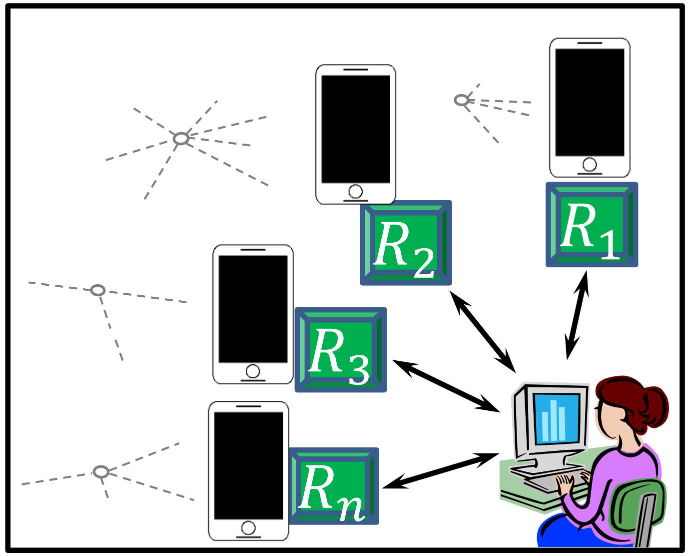
- Disadvantage of the local model:
  - data-thirsty (more data for the same accuracy)

Local Privacy Models with Graphs [Qin Yu Yang Khalil Xiao Ren 17]

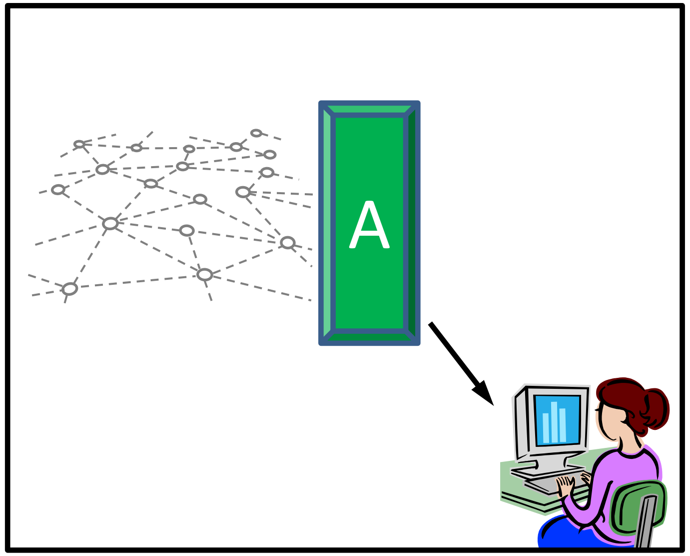
Local Noninteractive



Local (Interactive)



Centralized



- Each node in the graph represents a party
- Each party's input is the subgraph induced by the node and its neighbors

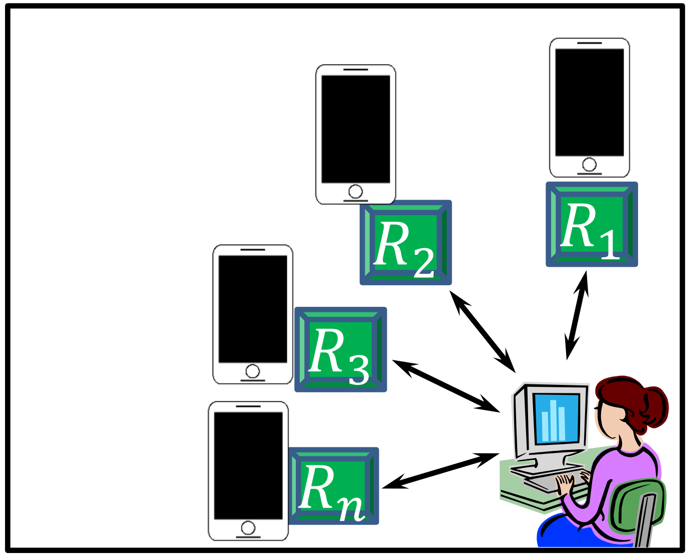
Note:
each edge is visible to two parties.

Conceptually different from the standard local model, where input is partitioned between parties

Prior Work on Local Graph Model

Empirical accuracy for subgraph counting and (informally-defined) synthetic graph generation

- [Qin, Yu, Yang, Khalil, Xiao, Ren. *CCS* 2017;
  Gao, Lil, Chen, Zou. *Trans. Comp. Soc. Sys.* 2018;
  Zhang, Wei, Zhang, Hu, Liu, *ICCNS* 2018;
  Sun, Xiao, Khalil, Yang, Qin, Wang, Yu, *CCS* 2019;
  Ye, Hu, Au, Meng, Xiao. *ICDE* 2020]



Theoretical guarantees for

- counting triangles, stars, 4-cycles

  [Imola, Murakami, Chaudhuri, *USENIX Security* 2021 and 2022, CCS 2022]

- other graph summaries (k-core decomposition, densest subgraphs)

  [Dhulipala, Liu, Raskhodnikova, Shi, Shun, Yu. *FOCS* 2022]

Results: Additive Error of Triangle Counting

- Triangle counting in the local model was first studied by [Imola Murakami Chaudhuri]

Model		Previous Results	Our Results
Noninteractive	Lower bounds	$\Omega\left(n^{3/2}\right)$ [IMC 21]	$\Omega(n^2)$
	Upper bounds	$O(n^2)$ (constant ϵ) [IMC 22b]	$O\left(\frac{n^2}{\epsilon} + \frac{n^{3/2}}{\epsilon^3}\right)$
Interactive	Lower Bounds	$\Omega(n)$ (easy)	$\Omega\left(\frac{n^{3/2}}{\epsilon}\right)$
	Upper bounds	$O\left(\frac{n^2}{\epsilon} + \frac{n^{3/2}}{\epsilon^2}\right)$ [IMC 22a]	

- Some upper bounds can also be expressed in terms of the number of 4-cycles

Randomized Response [Warner 63]

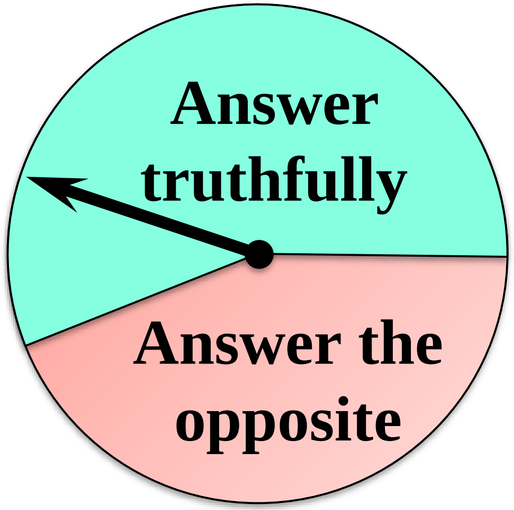
- Canonical example of a local algorithm

- Invented to help get truthful answers on sensitive YES/NO survey questions.

- Randomization operator takes $y \in \{0,1\}$:

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

ratio is e^ϵ



Triangle Counting Via Randomized Response

Triangle Count (**Input:** $\epsilon > 0,$ distributed $n \times n$ adjacency matrix $\mathbf{A}$)

1. For all $\{i,j\} \in \binom{[n]}{2}$, release $X_{\{i,j\}} \leftarrow RR_\epsilon(\boldsymbol{A_{ij}})$
2. For all $\{i,j\} \in \binom{[n]}{2}$, set $Y_{\{i,j\}} \leftarrow \frac{X_{\{i,j\}} \cdot (e^\epsilon + 1) - 1}{e^\epsilon - 1}$
3. For all $\{i,j,k\} \in \binom{[n]}{3}$, set $Z_{\{i,j,k\}} \leftarrow Y_{\{i,j\}} \cdot Y_{\{j,k\}} \cdot Y_{\{i,k\}}$
4. **Return** $\hat{T} \leftarrow \sum_{\{i,j,k\} \in \binom{[n]}{3}} Z_{\{i,j,k\}}$

Release each $\boldsymbol{A_{ij}}$ using randomized response

Normalized noisy edge variables so that $\mathbb{E}[Y_{\{i,j\}}] = \boldsymbol{A_{ij}}$

$\mathbb{E}[Z_{\{i,j,k\}}] = \boldsymbol{A_{ij}} \cdot \boldsymbol{A_{jk}} \cdot \boldsymbol{A_{ik}} = \mathbb{1}_{\{i,j,k\}} = \begin{cases} 1 & \text{if } \{i,j,k\} \text{ forms a triangle} \\ 0 & \text{otherwise} \end{cases}$

Return an unbiased estimate for the triangle count

- The variance of $\hat{T}$ is $O\left(\frac{n^4}{\epsilon^2} + \frac{n^3}{\epsilon^6}\right)$

Main Ideas Behind the $\Omega(n^2)$ Lower Bound

1. We will use a noninteractive local algorithm $\mathcal{A}$ for counting triangles with error $O(n^2)$ to mount a *reconstruction attack* in the central model.

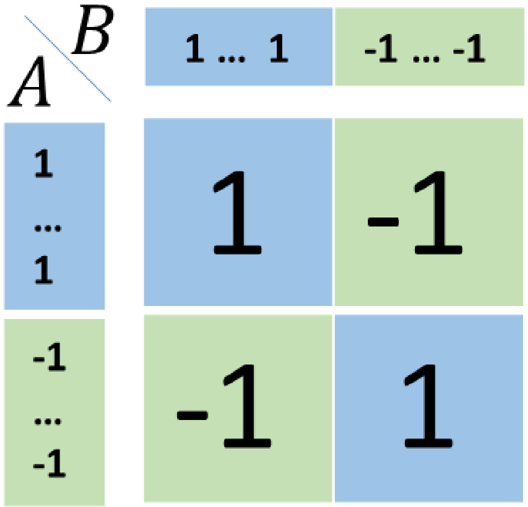
> Reconstruction attack [Dinur Nissim 03]
> *If an algorithm answers N random linear queries*
> *on a dataset of* $\mathbf{N}$ *bits*
> *with error* $\pm O(\sqrt{N})$
> *then a large constant fraction of the dataset can be reconstructed.*

2. Our dataset has $N = n^2$ bits, so we will answer (a constant fraction of) $\Theta(n^2)$ linear queries with error $\pm O(n)$.
3. To avoid invoking $\mathcal{A}$ separately for each query, we will develop a new type of linear queries called *outer-product* queries.
4. Instead of using $\mathcal{A}$ as a black box, we will used it as a ``gray box''

Outer-Product Queries

Let $X \in \{0,1\}^{n \times n}$ be a secret dataset (in the central model).

An outer-product query to X specifies two vectors A and B of length n with entries in $\{-1,1\}$ and returns $A^T X B$, that is, $\sum_{i,j \in [n]} A_i X_{ij} B_j$.

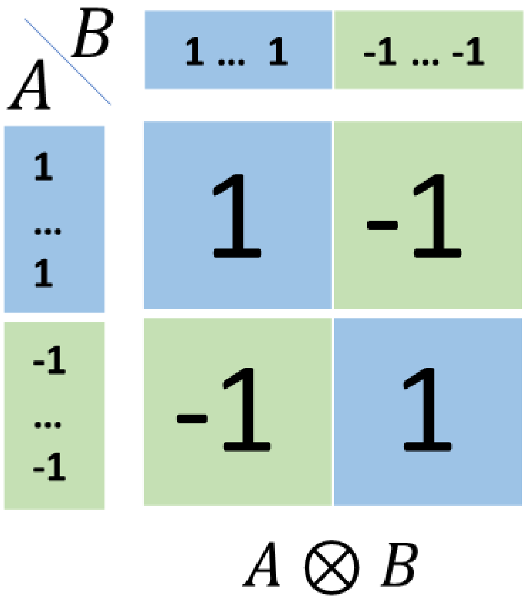


$A \otimes B$

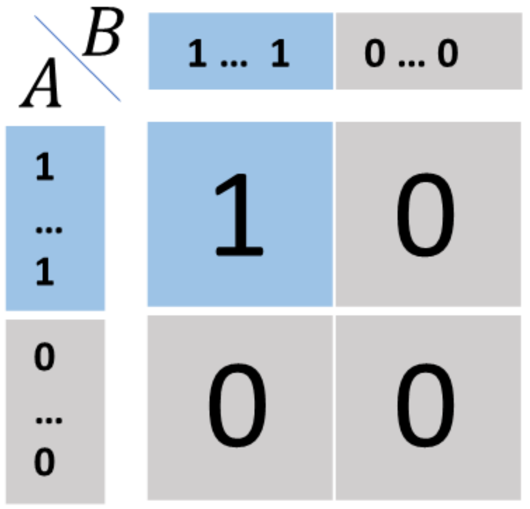
Outer-Product Queries vs. Submatrix Queries

Let $X \in \{0,1\}^{n \times n}$ be a secret dataset (in the central model).

An outer-product query to X specifies two vectors A and B of length n with entries in $\{-1,1\}$ and returns $A^T X B$, that is, $\sum_{i,j \in [n]} A_i X_{ij} B_j$.



A submatrix query is the same as an outer-product query, except that vectors A and B have entries in $\{0,1\}$ instead of $\{-1,1\}$.



Outer-Product Queries Can Be Simulated with Matrix Queires

$$
\underbrace{\begin{array}{c|cc} A \backslash B & 1 \dots 1 & -1 \dots -1 \\ \hline \begin{smallmatrix} 1 \\ \dots \\ 1 \end{smallmatrix} & 1 & -1 \\ \begin{smallmatrix} -1 \\ \dots \\ -1 \end{smallmatrix} & -1 & 1 \end{array}}_{A \otimes B}
= 2 \cdot \left(\underbrace{\begin{array}{c|cc} A' \backslash B' & 1 \dots 1 & 0 \dots 0 \\ \hline \begin{smallmatrix} 1 \\ \dots \\ 1 \end{smallmatrix} & 1 & 0 \\ \begin{smallmatrix} 0 \\ \dots \\ 0 \end{smallmatrix} & 0 & 0 \end{array}}_{A' \otimes B'}
+ \underbrace{\begin{array}{c|cc} A'' \backslash B'' & 0 \dots 0 & 1 \dots 1 \\ \hline \begin{smallmatrix} 0 \\ \dots \\ 0 \end{smallmatrix} & 0 & 0 \\ \begin{smallmatrix} 1 \\ \dots \\ 1 \end{smallmatrix} & 0 & 1 \end{array}}_{A'' \otimes B''} \right)
- \underbrace{\begin{array}{c|cc} \vec{1} \backslash \vec{1} & 1 \dots 1 & 1 \dots 1 \\ \hline \begin{smallmatrix} 1 \\ \dots \\ 1 \end{smallmatrix} & 1 & 1 \\ \begin{smallmatrix} 1 \\ \dots \\ 1 \end{smallmatrix} & 1 & 1 \end{array}}_{\vec{1} \otimes \vec{1}}
$$

Main Lemma

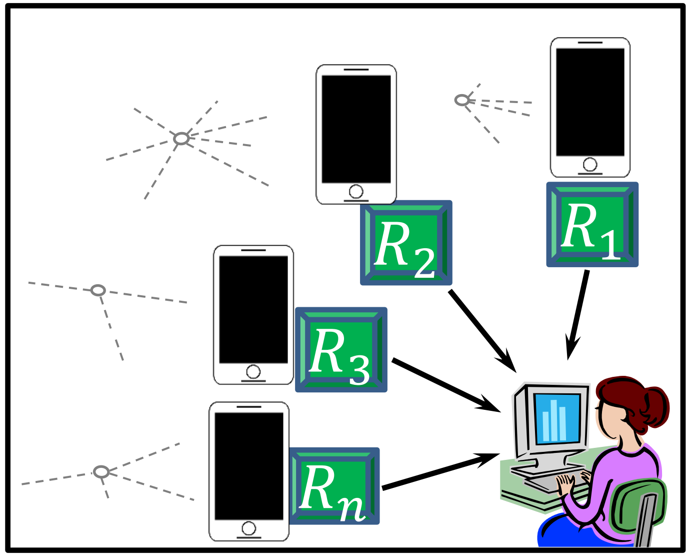
Answering Outer-product Queries via Triangle Counting

Suppose there is a *noninteractive local* (ϵ,δ)-DP algorithm $\mathcal{A}$ that,
for every $3n$-node graph, with probability $\Omega(1)$ returns the number of triangles $\pm O(n^2)$.

$\Downarrow$

Then there is a $(2\epsilon, 2\delta)$-DP algorithm $\mathcal{B}$ in the *central model* that,
for every secret dataset $X \in \{0,1\}^{n\times n}$ and every set of k outer-product queries,
with probability $\Omega(1)$ returns a vector of answers, $\Omega(k)$ of which have error $\pm O(n)$.

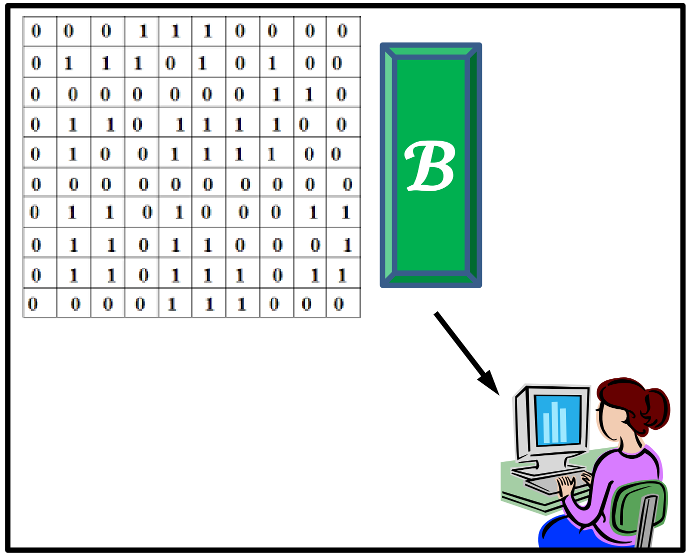
- Note: algorithm $\mathcal{A}$ is specified by
  - a local randomizer R_i for each vertex i
  - a postprocessing algorithm $\mathcal{P}$



Construction of Algorithm $\mathcal{B}$

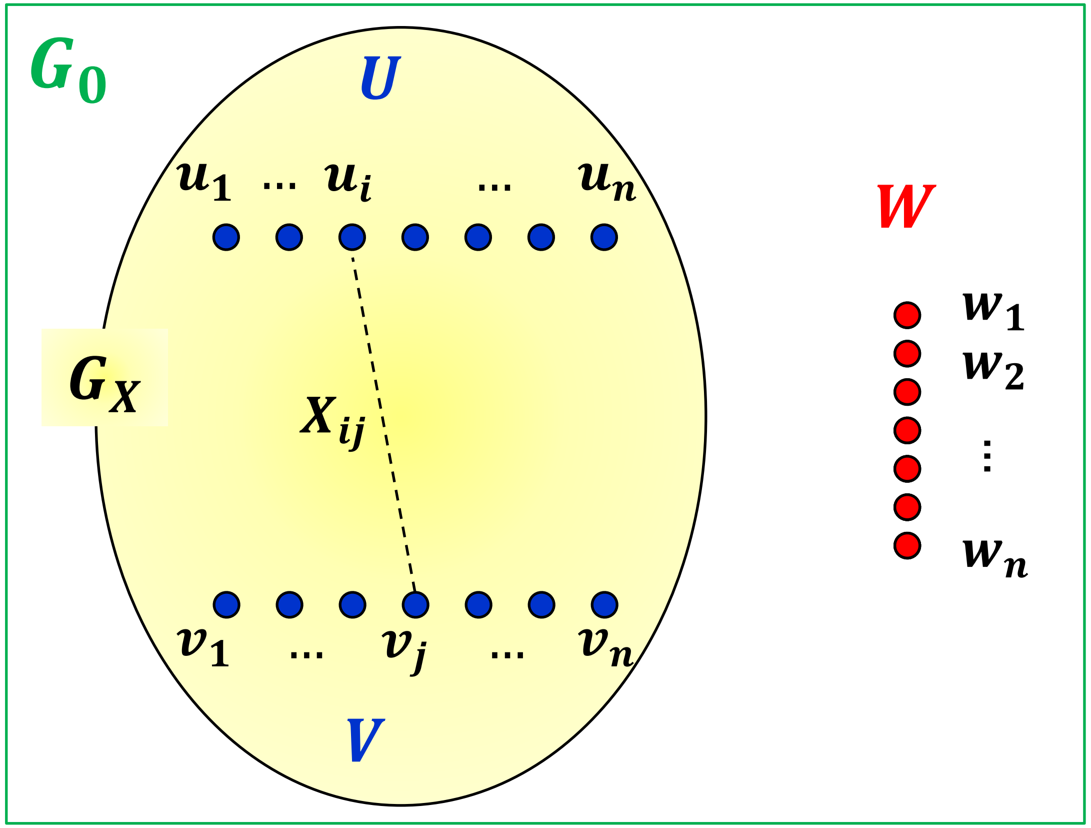
- Algorithm $\mathcal{B}$ converts its input dataset dataset $X \in \{0,1\}^{n\times n}$ to two graphs, G_0 and G_1, and runs local randomizers on them.
- After that, $\mathcal{B}$ does not touch X.
- It simulates outer-product queries with matrix queries
- For each matrix query (A, B), algorithm $\mathcal{B}$
  - constructs a query graph $G_{(A,B)}$,
  - estimates the number of triangles in $G_{(A,B)}$ by *mixing and matching* the responses of the local randomizers on G_0 and G_1 and running the postprocessing algorithm $\mathcal{P}$ on them,
  - uses the result to answer the query.

Centralized



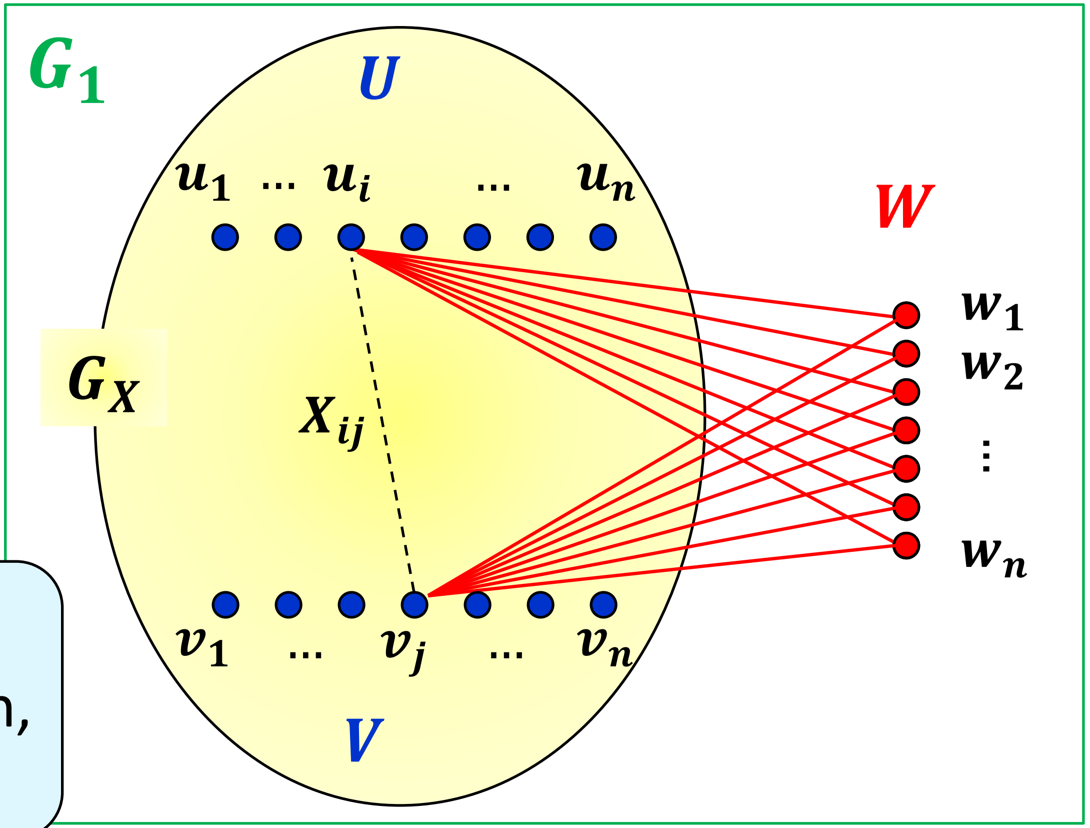
Construction of Graphs G_0 and G_1 from Dataset X

- All graphs will be on $3n$ nodes
- Create 3 sets U, V, W with n nodes in each
- Create a *secret* bipartite subgraph G_X on (U, V) with edges determined by dataset X
- The resulting graph is G_0



Construction of Graphs G_0 and G_1 from Dataset X

- All graphs will be on $3n$ nodes
- Create 3 sets U, V, W with n nodes in each
- Create a *secret* bipartite subgraph G_X on (U, V) with edges determined by dataset X
- The resulting graph is $\boldsymbol{G_0}$
- For $\mathbf{G_1}$: add a complete bipartite graph between $\boldsymbol{U} \cup \boldsymbol{V}$ and $\boldsymbol{W}$



Algorithm $\mathcal{B}$ creates G_0 and G_1 from X,
runs local randomizer R_v for each vertex v for both,
and records the answers as $r_0(v)$ and $r_1(v)$

$\mathcal{B}$ won't touch X after this, so by composition $\mathcal{B}$ is $(2\epsilon, 2\delta)$-DP

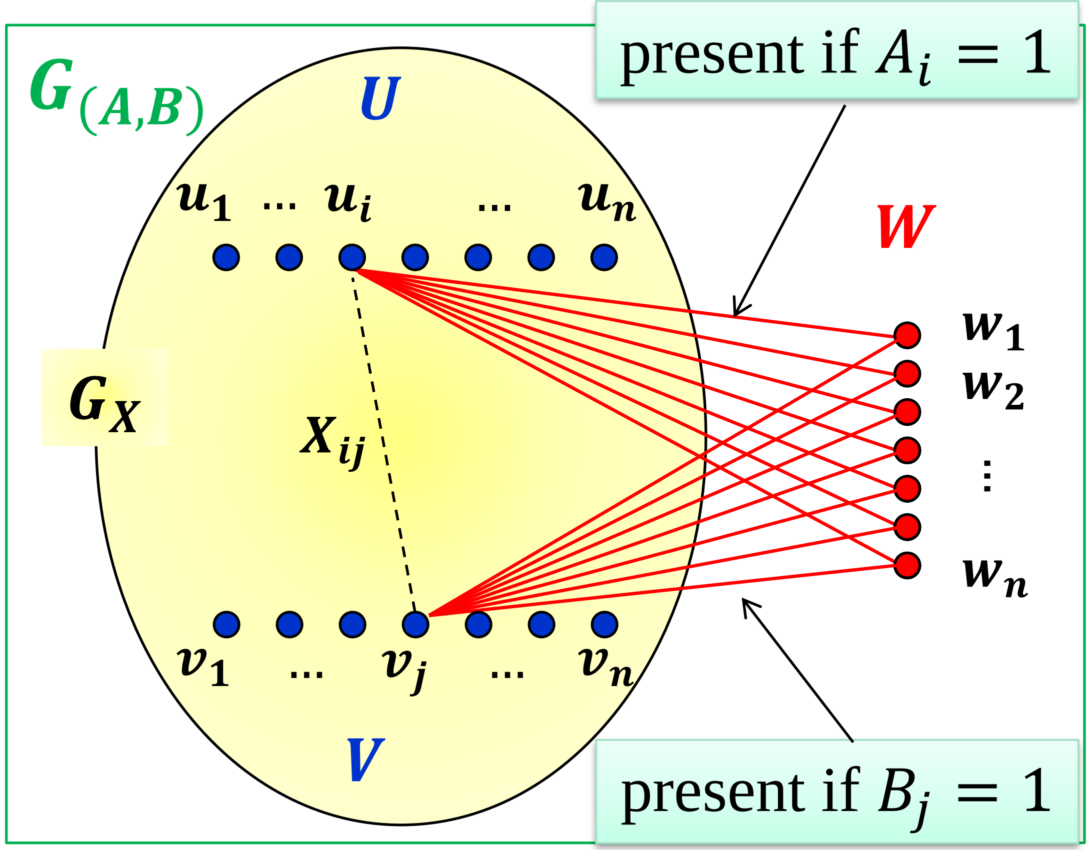
Construction of Query Graph $G_{(A,B)}$ *for Matrix Query* $(\boldsymbol{A},\boldsymbol{B})$

- Start with G_0
- Each node $u_i \in U$ connects to all nodes in W iff $A_i = 1$
- Each node $v_j \in V$ connects to all nodes in W iff $B_j = 1$

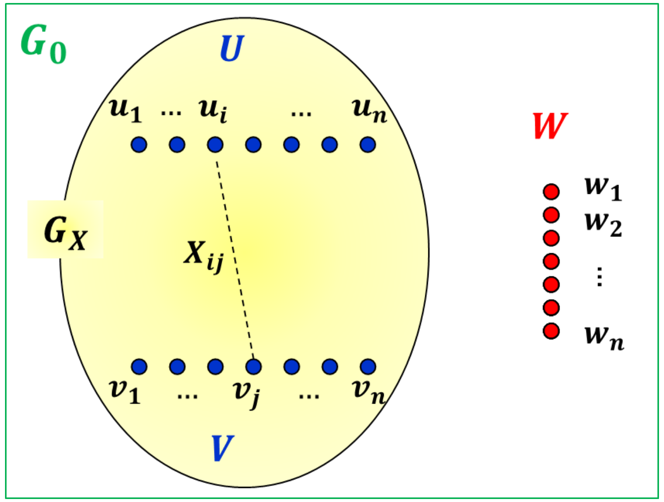
Each pair (u_i, v_j) contributes n triangles if $X_{ij} = A_i = B_j = 1$, and no triangles otherwise.

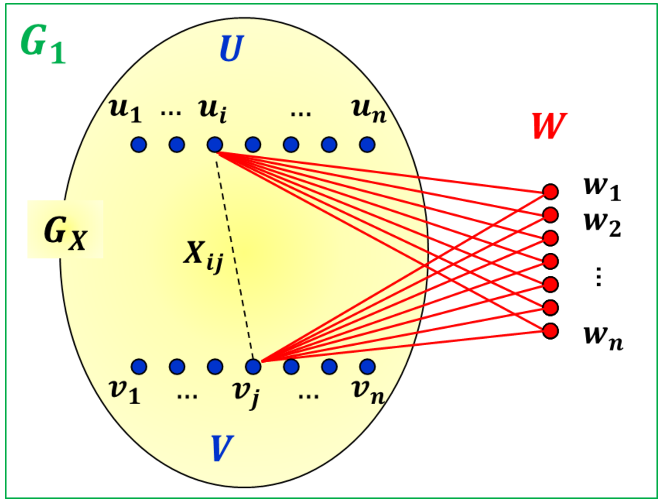
The number of triangles in $G_{(A,B)}$ is

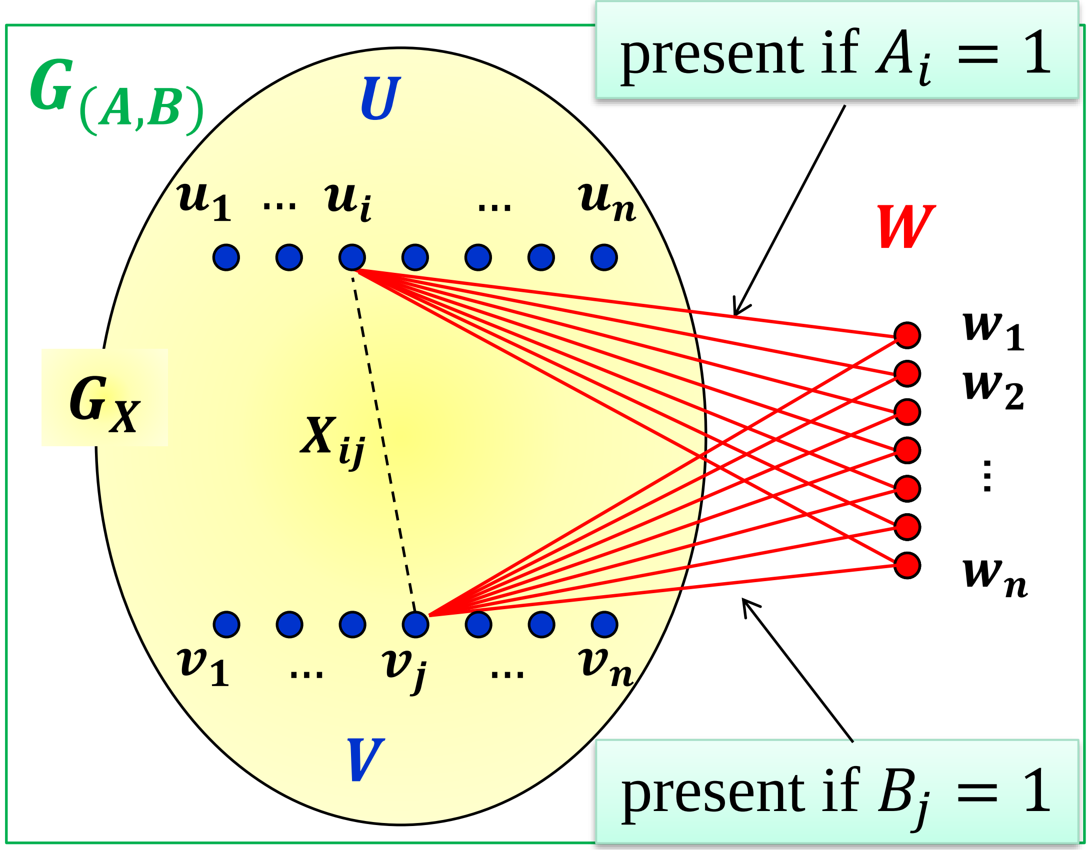
$$\sum_{i,j\in[n]} n\, A_i X_{ij} B_j = n \cdot A^T X B$$



Mix-and-Match Strategy to Simulate $\mathcal{A}$ on Query Graph $G_{(A,B)}$







- For all $u_i \in U$: $\text{view}_{u_i}(G_{(A,B)}) = \text{view}_{u_i}(G_{A_i})$
- For all $v_j \in V$: $\text{view}_{v_j}(G_{(A,B)}) = \text{view}_{v_j}(G_{B_j})$

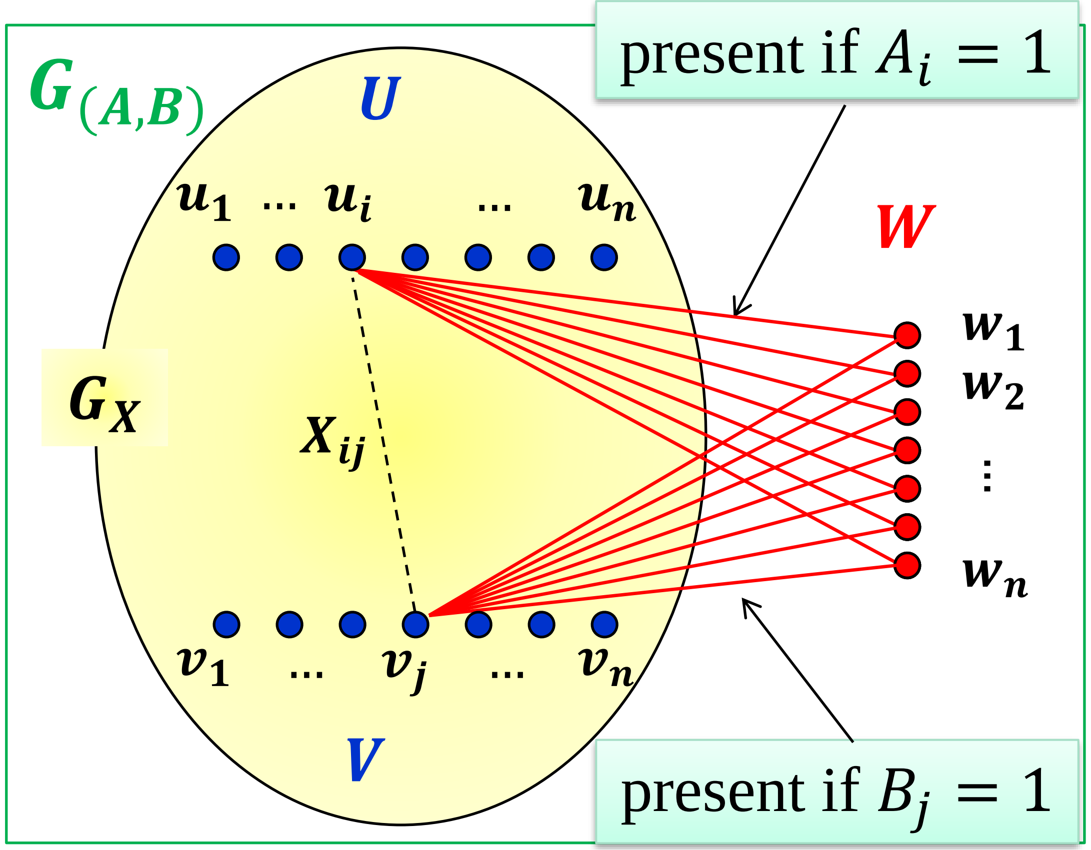
Algorithm $\mathcal{B}$ already ran the local randomizer for both possible views for all nodes v in $U \cup V$ and recorded the answers as $r_0(v)$ and $r_1(v)$

Other nodes do not have access to secret dataset X

Answering Most Matrix Queries Accurately

$\mathcal{B}$ runs the triangle-counting algorithm as a gray box by mixing and matching the recorded answers $r_0(v)$ and $r_1(v)$ for different nodes

- If the triangle-counting algorithm has error $\pm O(n^2)$, then $\mathcal{B}$ can answer submatrix queries with error $\pm O(n)$.
- The expected number of queries answered inaccurately is small.
- Markov inequality guarantees that most are answered accurately (with sufficient probability).



We Proved the Main Lemma

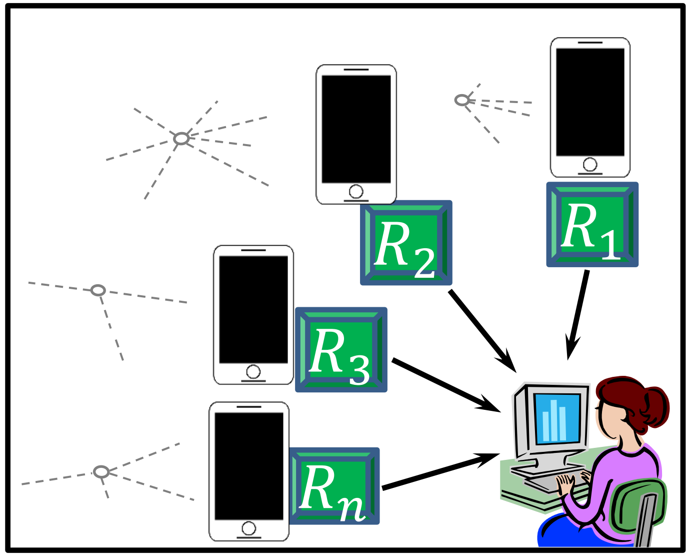
Answering Outer-product Queries via Triangle Counting

Suppose there is a *noninteractive local* (ϵ, δ)-DP algorithm $\boldsymbol{\mathcal{A}}$ that,

for every $3n$-node graph, with probability $\Omega(1)$ returns the number of triangles $\pm O(n^2)$.

$\Downarrow$

Then there is a $(2\epsilon, 2\delta)$-DP algorithm $\boldsymbol{\mathcal{B}}$ in the *central model* that,
for every secret dataset $X \in \{0,1\}^{n \times n}$ and every set of k outer-product queries,
with probability $\Omega(1)$ returns a vector of answers, $\Omega(k)$ of which have error $\pm O(n)$.



Anti-Concentration for Random Outer-Product Queries

Anti-Concentration Theorem

Let M be an $n \times n$ matrix with entries in $\{-1,0,1\}$ and m be the number of nonzero entries in M.

Let A and B be drawn u.i.r. from $\{-1,1\}^n$.

If $m \geq \gamma\, n^2$ for some constant γ, then

$$\Pr\left[|A^T M\, B| > \frac{\sqrt{m}}{2}\right] \geq \frac{\gamma^2}{16}.$$

Think of M as $X - Y$, where X is the dataset and Y is potential reconstruction

i.e., the number of entries on which X and Y differ

W.h.p., the outer-product query (A, B) gives sufficiently different answers on X and Y to rule out Y.

Anti-Concentration for Random Outer-Product Queries

Anti-Concentration Theorem

Let M be an $n \times n$ matrix with entries in $\{-1,0,1\}$ and m be the number of nonzero entries in M. Let A and B be drawn u.i.r. from $\{-1,1\}^n$. If $m \geq \gamma\, n^2$ for some constant γ, then

$$\Pr\left[|A^T M\, B| > \frac{\sqrt{m}}{2}\right] \geq \frac{\gamma^2}{16}.$$

Understanding individual query entries

Let $Z_{ij} = A_i B_j$ for $i,j \in [n]$

by independence of A_i and B_j

$\mathbb{E}(Z_{ij}) = \mathbb{E}(A_i) \cdot \mathbb{E}(B_j) = 0$

$\mathrm{Var}(Z_{ij}) = \mathbb{E}(Z_{ij}^2) = \mathbb{E}(A_i^2 \cdot B_j^2) = 1$

Let $W = A^T M\, B$

- $\mathbb{E}(W) = \mathbb{E}\left(\sum_{i,j\in[n]} M_{ij} Z_{ij}\right) = \sum_{i,j\in[n]} M_{ij}\mathbb{E}(Z_{ij}) = 0$

by pairwise independence of Z_{ij}

- $\mathrm{Var}(W) = \mathrm{Var}\left(\sum_{i,j\in[n]} M_{ij} Z_{ij}\right) = \sum_{i,j\in[n]} M_{ij}^2\, \mathrm{Var}(Z_{ij}) = \sum_{i,j\in[n]} M_{ij}^2 = m$

The theorem is proved by analyzing $\mathbb{E}(W^4)$

The Reconstruction Attack with Outer-Product Queries

Attacker (**Input:** dataset $\mathbf{X} \in \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{n} \times \boldsymbol{n}}$)

1. Select $k = \Theta(n^2)$ outer-product queries uniformly at random
2. Run algorithm $\boldsymbol{\mathcal{B}}$ on dataset X and the outer-product queries
3. Call an answer a to a linear query Q inaccurate on dataset Y if $|Q \cdot Y - a| > \frac{n}{12}$
4. **Return** any dataset Y^* on which at most $\frac{k}{6^4}$ answers are inaccurate

- When algorithm $\boldsymbol{\mathcal{B}}$ returns accurate answers, dataset X satisfies the requirement, so the attack will output a candidate dataset.
- By the Anti-Concentration Theorem and Chernoff bound, all datasets that differ from X on at least $1/9$ fraction of the entries are ruled out w.h.p.
- The attack succeeds w.h.p., so an accurate local DP-algorithm for triangle-counting does not exist.

Results: Additive Error of Triangle Counting

- Triangle counting in the local model was first studied by [Imola Murakami Chaudhuri]

Model		Previous Results	Our Results
Noninteractive	Lower bounds	$\Omega\left(n^{3/2}\right)$ [IMC 21]	$\Omega(n^2)$
	Upper bounds	$O(n^2)$ (constant ϵ) [IMC 22b]	$O\left(\frac{n^2}{\epsilon}+\frac{n^{3/2}}{\epsilon^3}\right)$
Interactive	Lower Bounds	$\Omega(n)$ (easy)	$\Omega\left(\frac{n^{3/2}}{\epsilon}\right)$
	Upper bounds	$O\left(\frac{n^2}{\epsilon}+\frac{n^{3/2}}{\epsilon^2}\right)$ [IMC 22a]	

Proved by a black-box reduction from computing summation of n bits in the local model.
Summation has additive error $\Omega(\sqrt{n}/\epsilon)$ [Joseph Mao Neel Roth 19]

Summary

- Improved bounds for triangle-counting in the local model
  - Tight bounds in terms of the number of nodes, n, for the noninteractive model
- Developed techniques for proving lower bounds for graph problems in the local model
  - Use of reconstruction attacks in the local model
  - New type of linear queries (outer-product queries)
  - mix-and-match strategy that runs the local randomizers with different completions of their adjacency lists

Open Questions

- Tight bounds for triangle counting in the local interactive model?
- Better understanding of graph analysis in the local model with edge-DP and node-DP
- What local models make sense in terms of privacy and distribution of input?