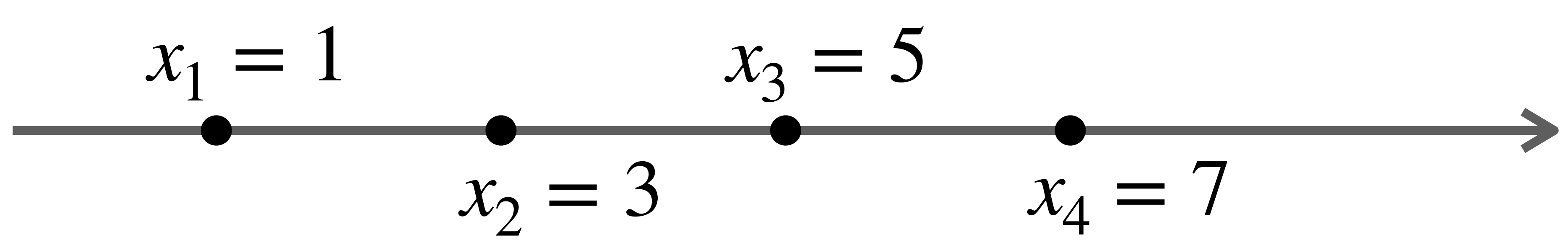
Differentially Private Range Query on Shortest Paths

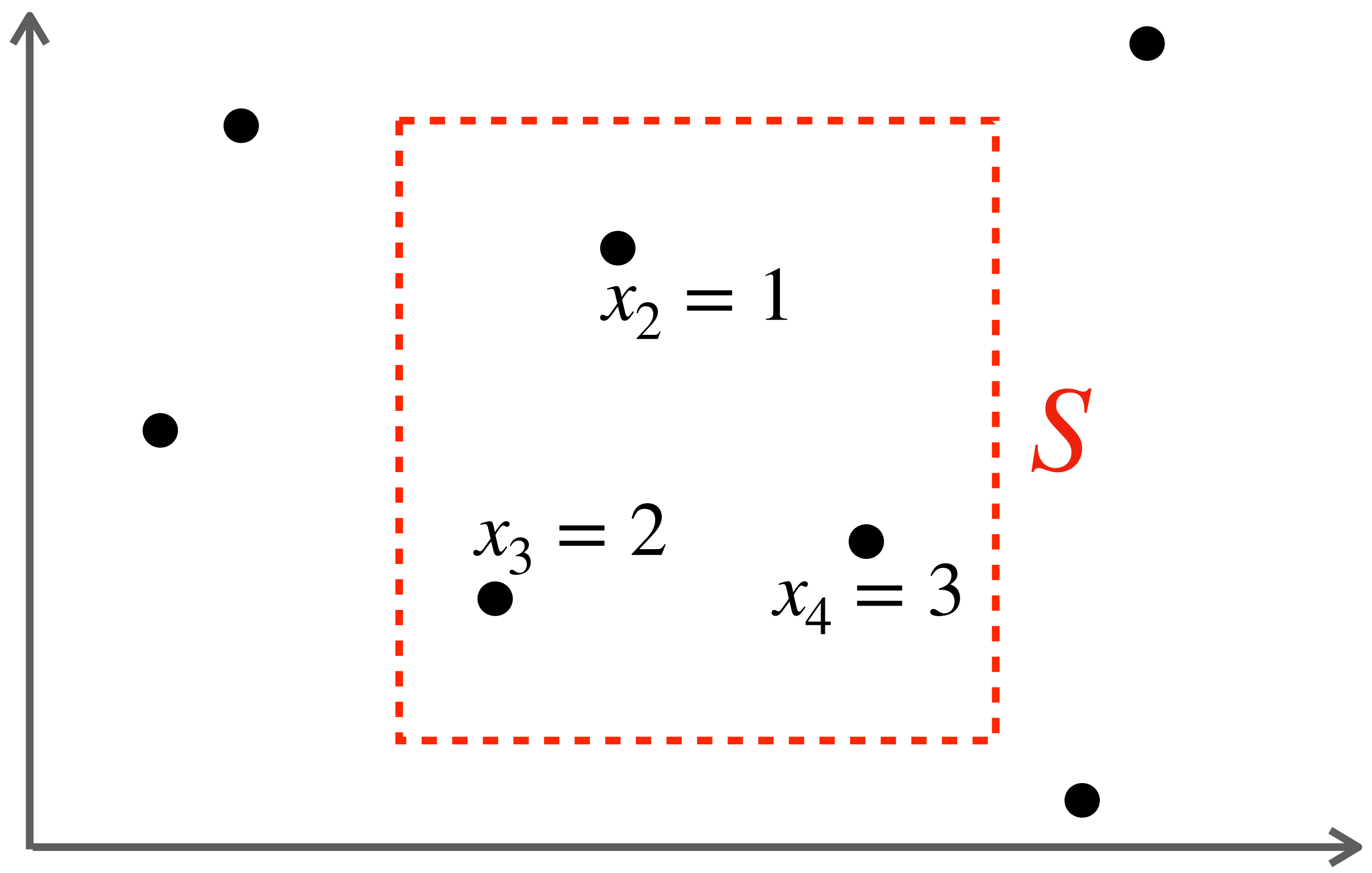
Chengyuan Deng
Rutgers University

Joint work with Jie Gao, Jalaj Upadhyay, Chen Wang

Range Query

- Given a set $X = \{x_1, \ldots, x_n\}$ and a query function, range query $q_f(X, i, j)$ returns $f(X[i, j]) = f(x_i, \ldots, x_j)$
- Geometric range query: (i, j) is the range of a geometric shape





Example: line

- Query function $f = \Sigma$
- $q_f(X,2,3) = x_2 + x_3 = 8$

Example: Square

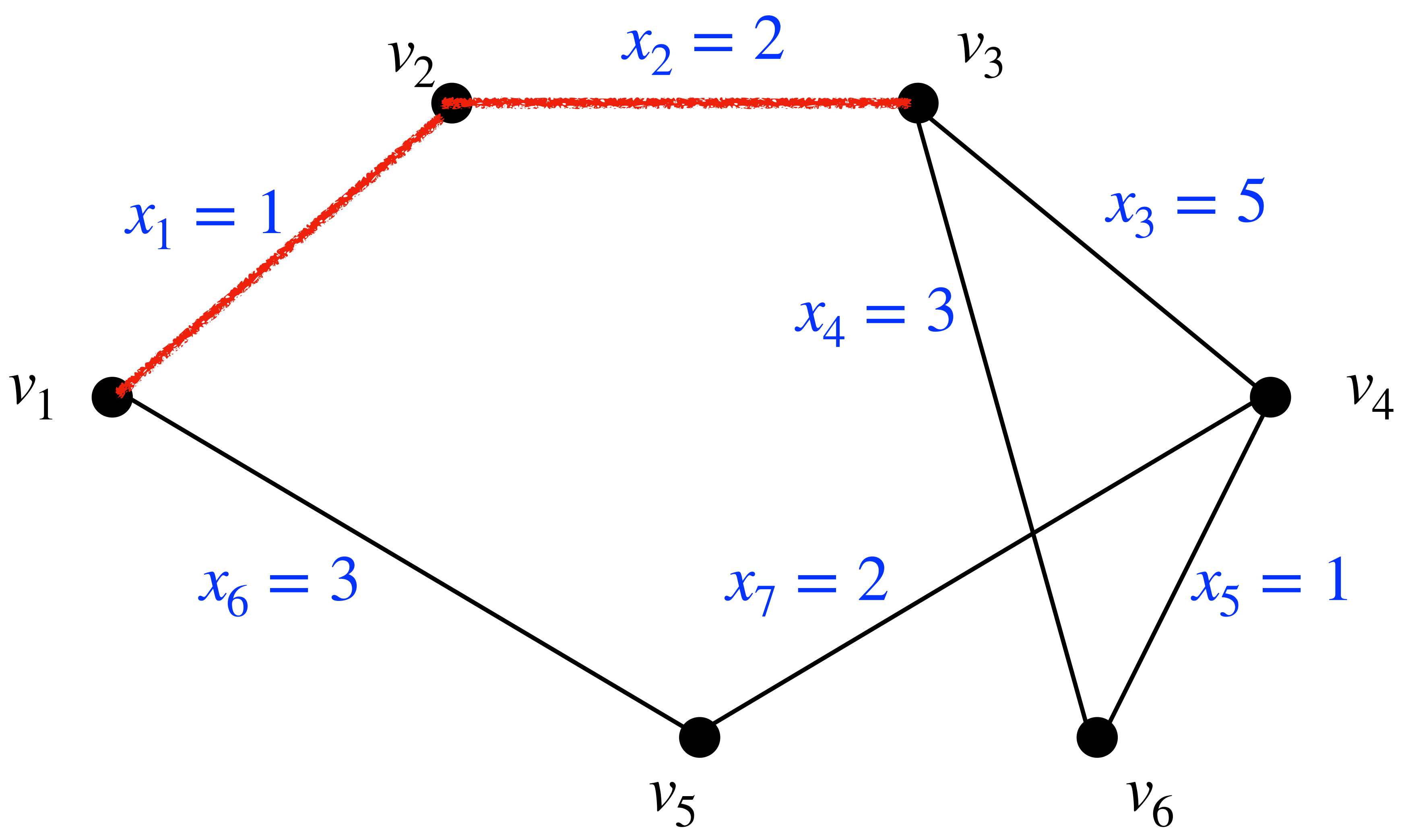
- Query function $f = \max$
- $q_f(X, S) = \max(x_2, x_3, x_4) = 3$

Range Query on Graphs

- Range Query has been widely studied with geometric ranges

What if the range is non-geometric?

- Motivating scenario



Example: graph

- $G = (V, E, X)$ where X is the set of edge attributes
- Query function $f = \Sigma$
- $q_f(X, v_1, v_3) = x_1 + x_2 = 3$

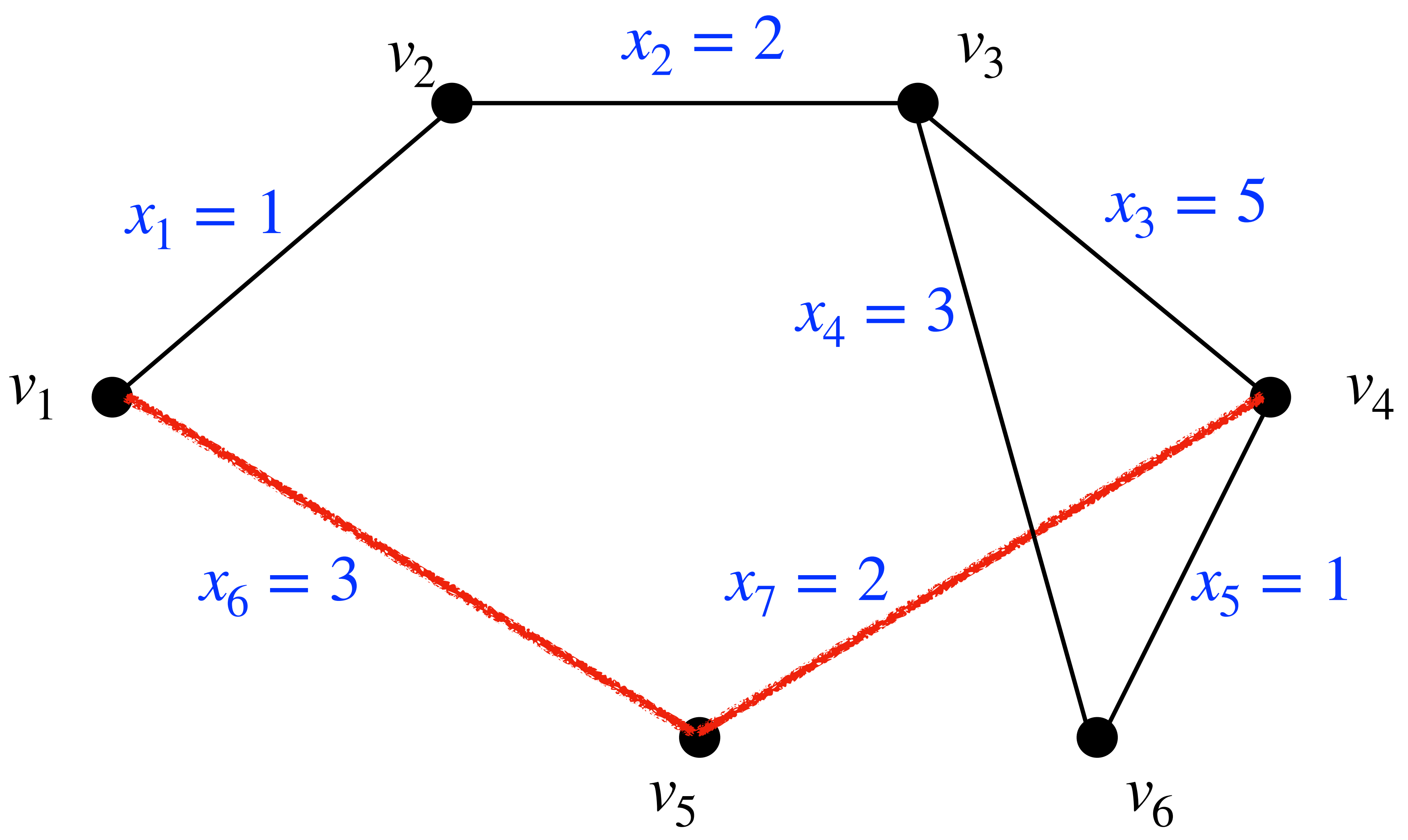
- Note: Throughout, we have notation $|V| = n, |E| = m$

Range Query on Graphs

- Range Query has been widely studied with geometric ranges

What if the range is non-geometric?

- Motivating scenario



Example: shortest paths

- For $q_f(X, v_i, v_j)$, take the edges along the shortest path between (v_i, v_j)
- Query function $f = \Sigma$
- $q_f(X, v_1, v_4) = x_6 + x_7 = 5$

If the edge attribute is the edge weight, then $q_f(X, v_i, v_j)$ returns the shortest distance between (v_i, v_j)

All Sets Range Query on Shortest Paths

- Given $G = (V, E, X)$ and a query function f, the All Sets Range Query (ASRQ) on G returns $q_f(X, v_i, v_j)$ for all pairs of $(v_i, v_j) \in V \times V$

- If $X(v_i, v_j) = d(v_i, v_j)$, then ASRQ is equivalent to All pairs shortest distances (APSD)

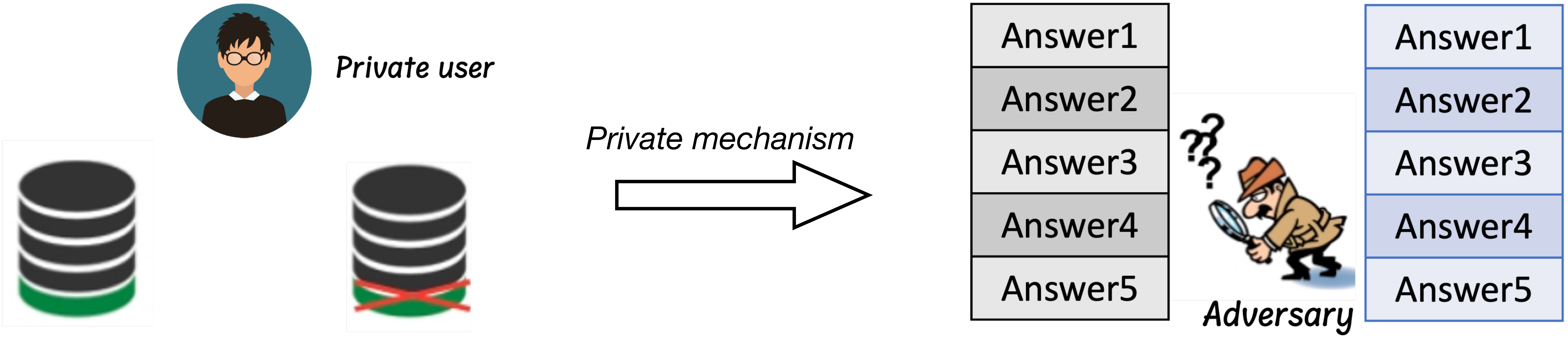
Application

- Financial security in trading networks: combat fraud
- Analysis in supply chain networks: end-to-end resilience, etc.

Our goal: Protect the sensitive information in networks via differential privacy, with the smallest possible error.

Differential Privacy

- Key idea: Protect the data such that for neighboring datasets differed by only one instance, the adversary cannot distinguish the outcome.

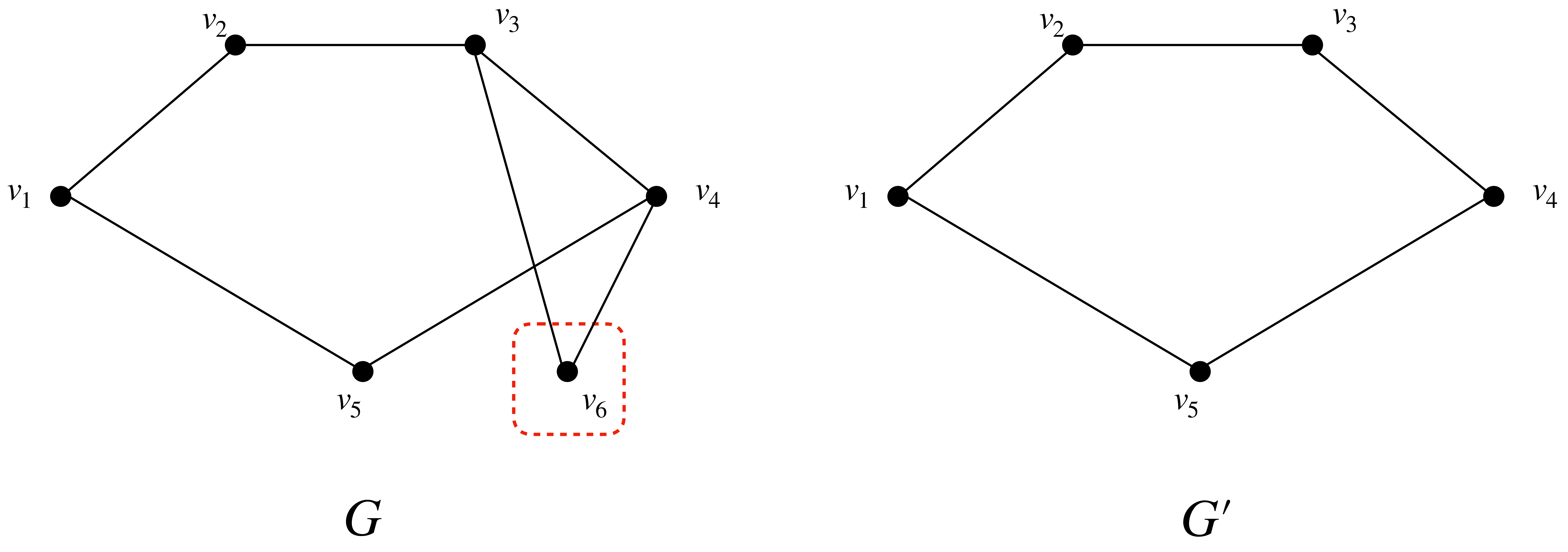


- Compromise: some error in reported answers

Next: define neighboring graphs for our setting

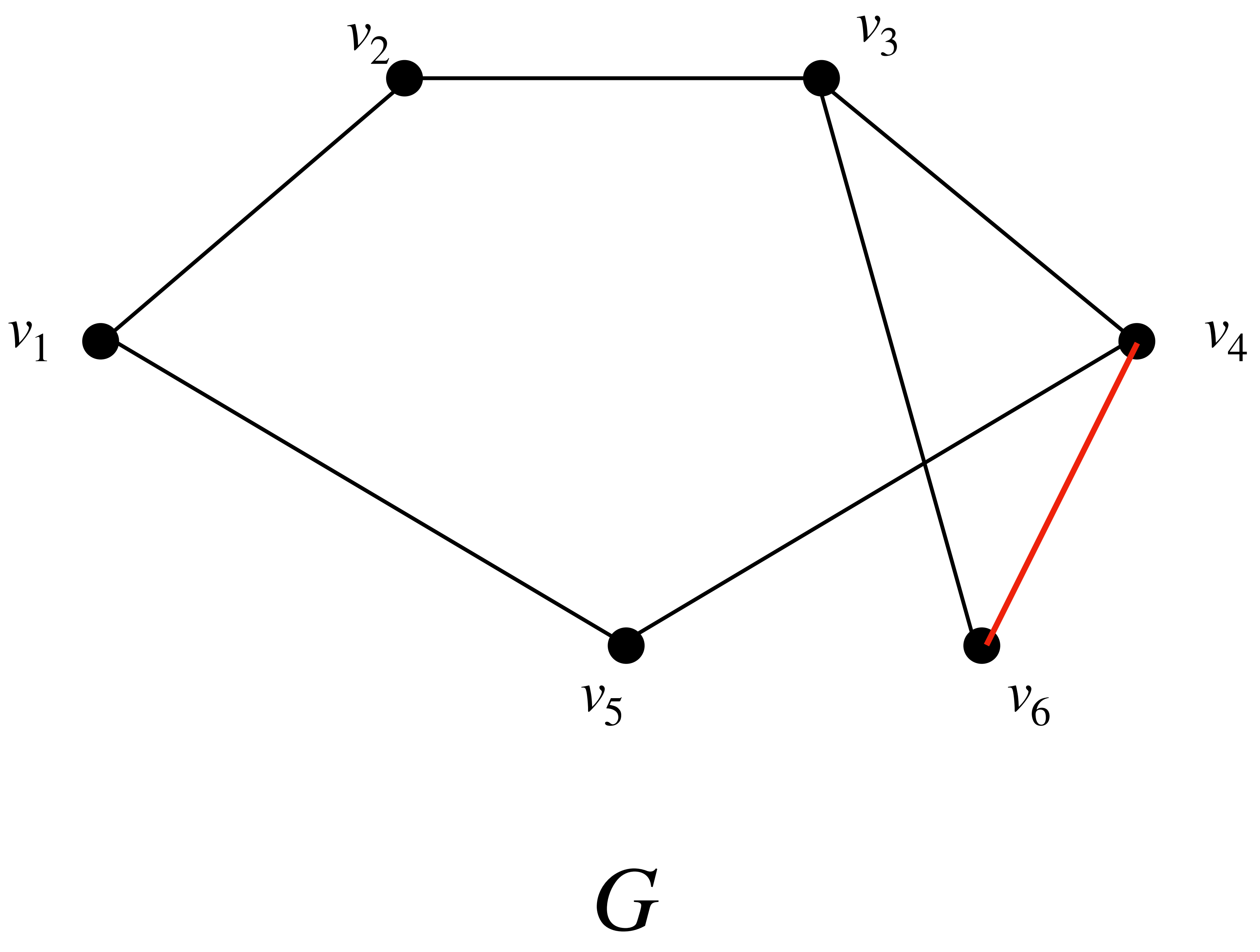
Differential Privacy for Graph

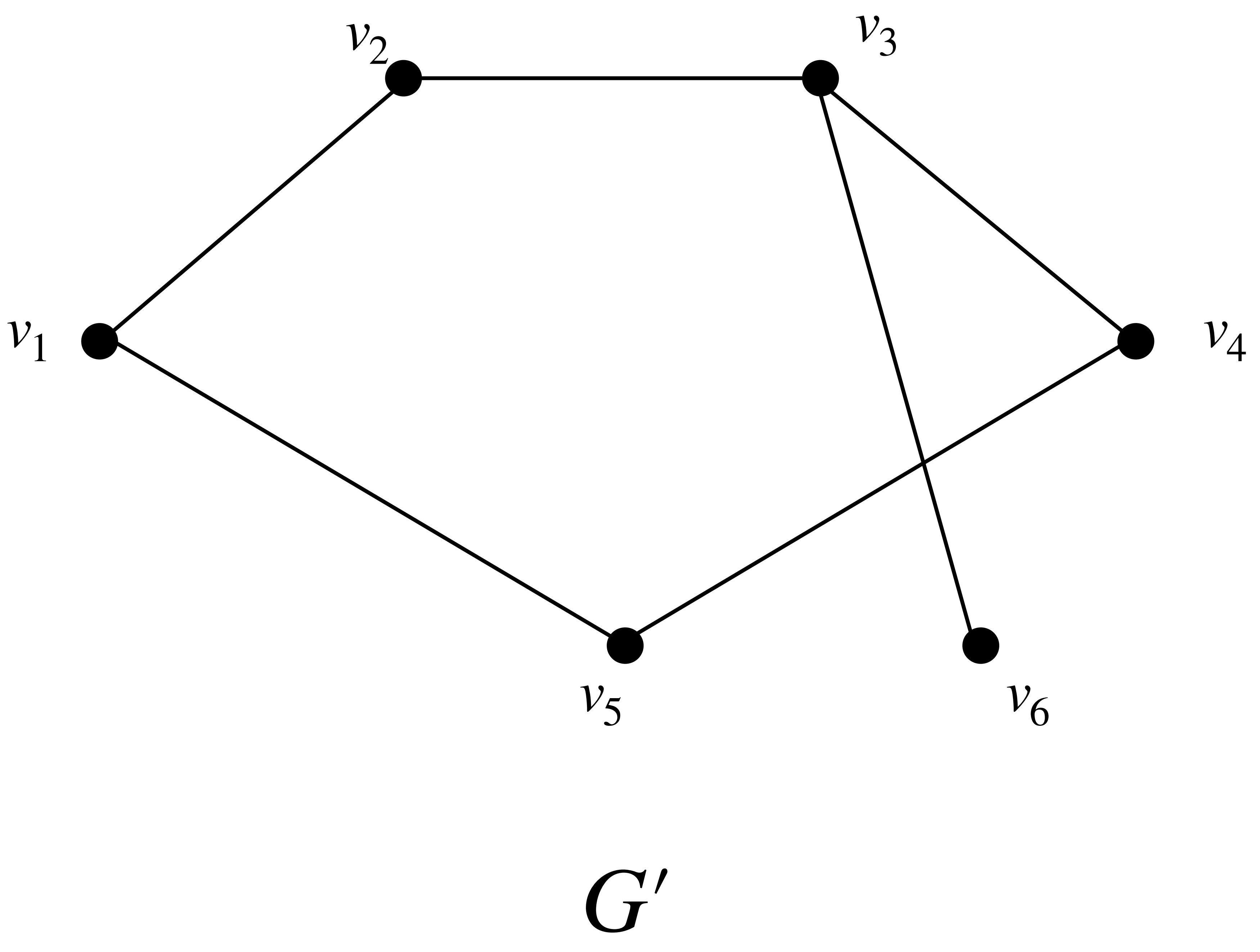
- Three notions of neighboring graphs have been proposed:
  - Node-level privacy



Differential Privacy for Graph

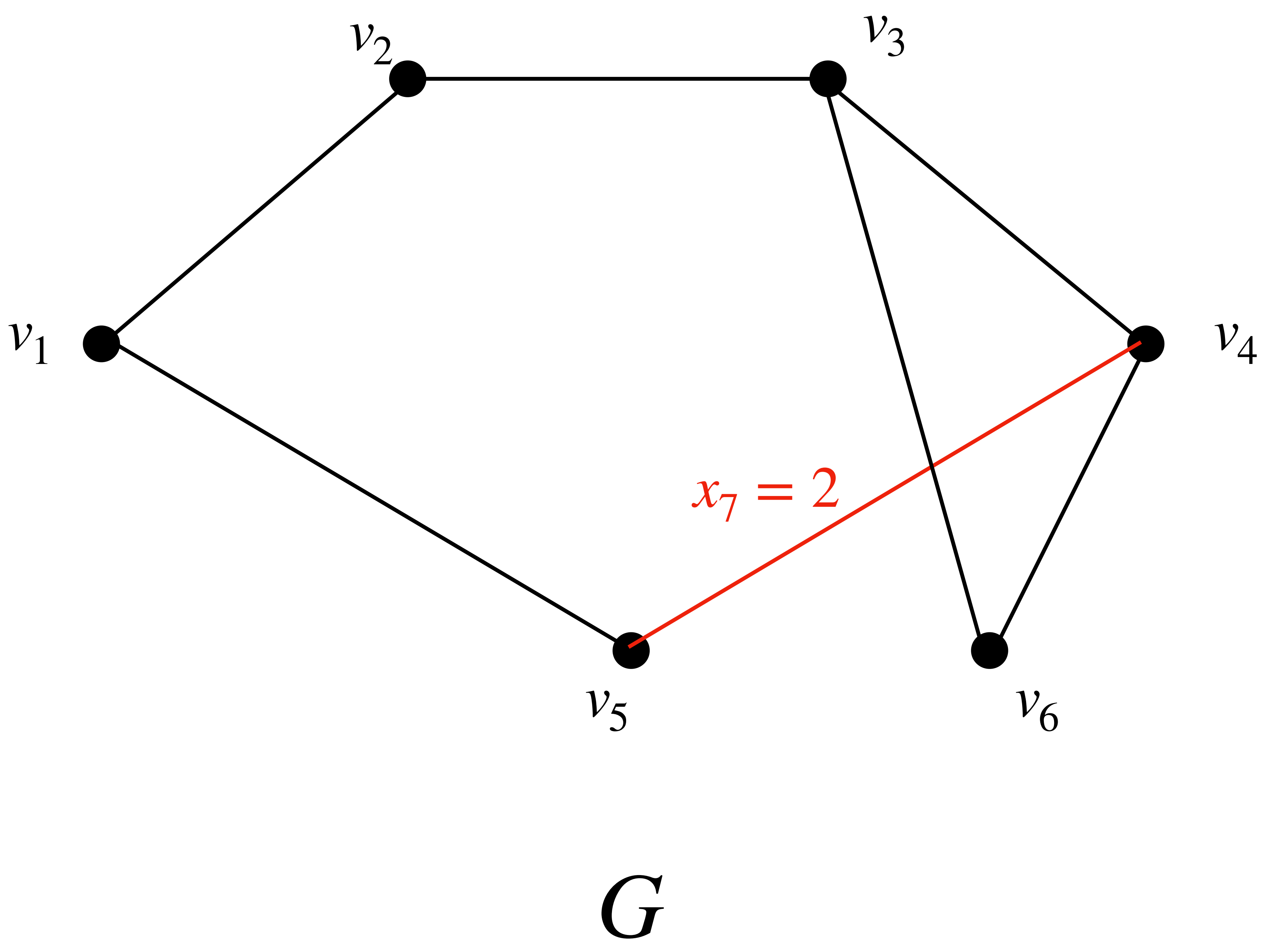
- Three notions of neighboring graphs have been proposed:
  - Node-level privacy
  - Edge-level privacy

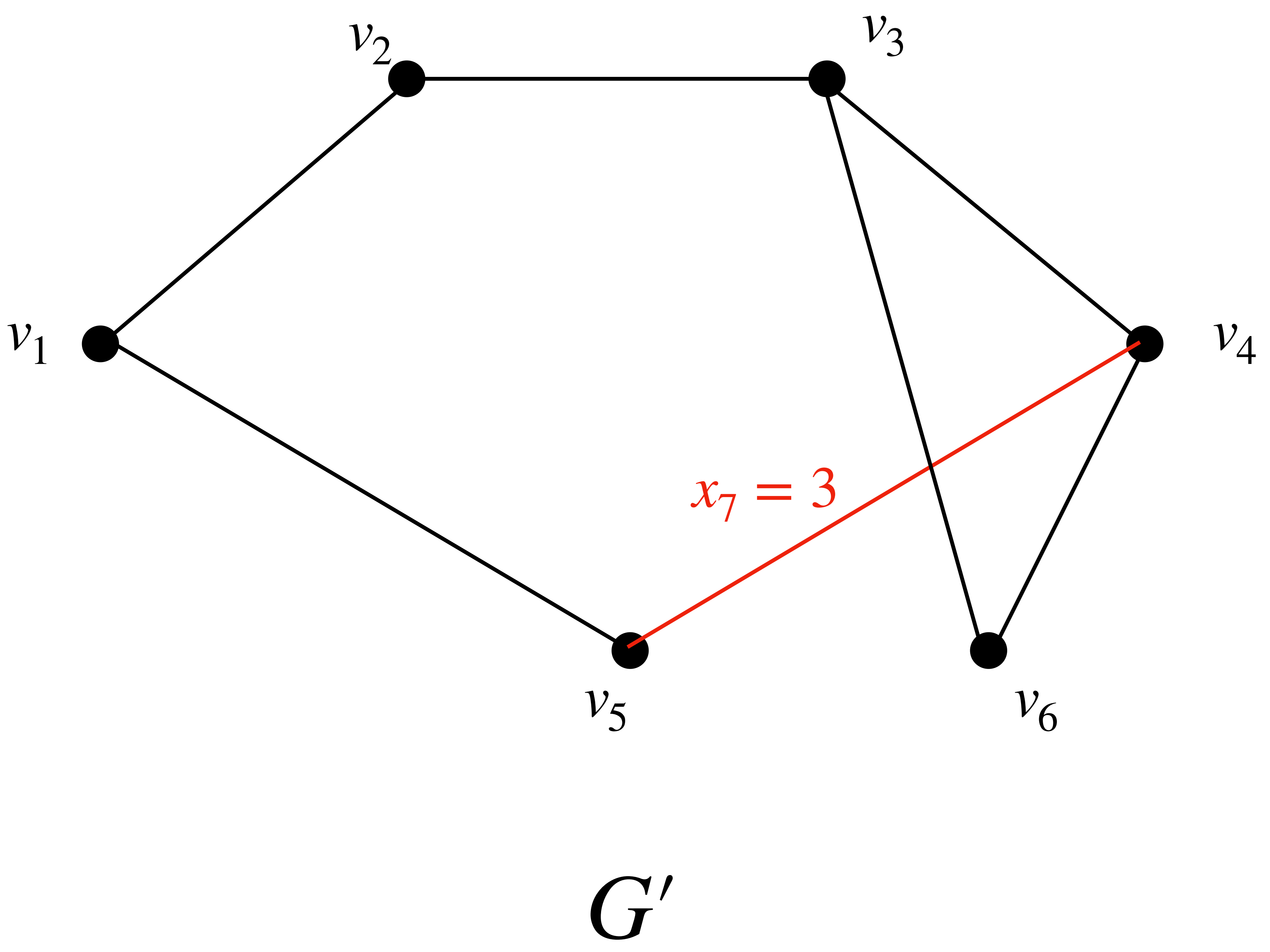




Differential Privacy for Graph

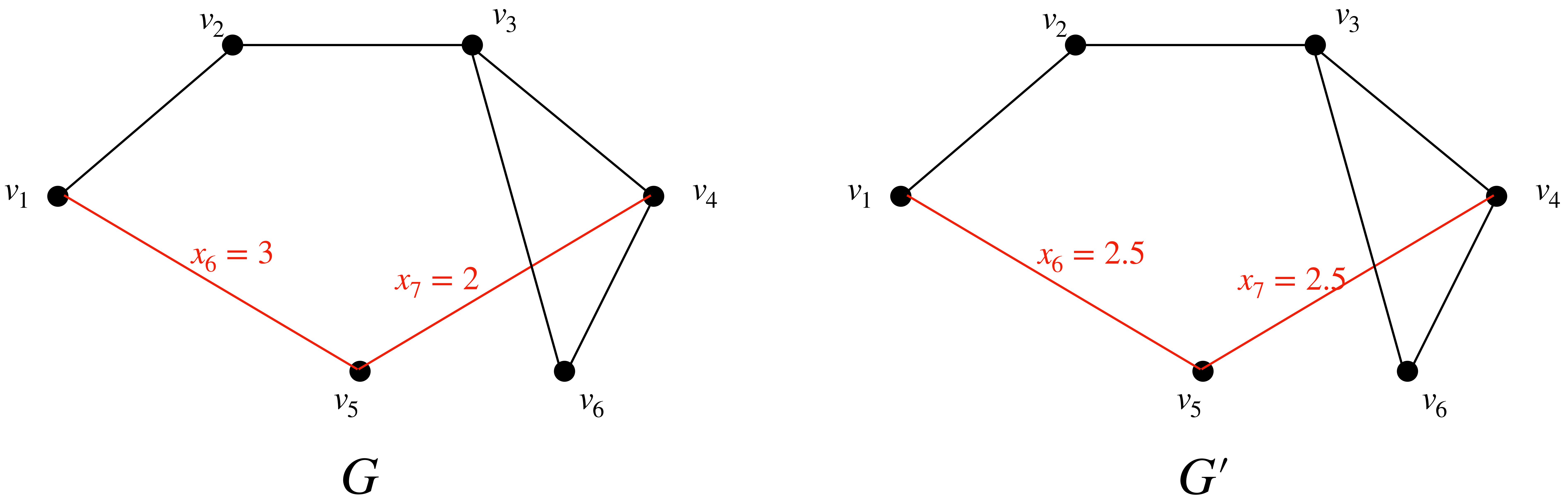
- Three notions of neighboring graphs have been proposed:
  - Node-level privacy
  - Edge-level privacy
  - Weight-level privacy [Sealfon '16]





Differential Privacy for Graph

- We choose weight-level privacy on attributes as our privacy model
    - The edge attribute is the sensitive information we want to protect
    - The weight-level privacy is proposed to study the DP-APSD problem
- Definition: let $w, w' : X \to \mathbb{R}^{\geq 0}$ be functions that map each element in X to a non-negative real number, we say w, w' are neighboring if $\sum_{x \in X} |w(x) - w'(x)| \leq 1.$



Formalize the Problem — DP-ASRQ

- We consider two query functions:
  - Counting query： $f = \Sigma$
  - Bottleneck query： $f = \max / \min$
- Differentially Private All Sets Range Query

Let $(R(X, S), f)$ be a range query system and w, w' be any neighboring attribute functions. An algorithm $\mathscr{A}$ is (ε, δ)-DP if for all sets of possible outputs C, we have:

$$\Pr[\mathscr{A}(R, f, w) \in C] \leq e^{\varepsilon} \cdot \Pr[\mathscr{A}(R, f, w') \in C] + \delta\,.$$

$\delta = 0 \longrightarrow \varepsilon$-DP

- Goal: minimize the additive error

$$\min \max_{u,v \in V} |\mathscr{A}(f(u, v) - f(u, v)|$$

Our results

	Pure DP	Approximate DP	Lower bound
Counting	$\tilde{O}(\frac{n^{1/3}}{\varepsilon})$	$\tilde{O}(\frac{n^{1/4}\log^{1/2} 1/\delta}{\varepsilon})$	$\tilde{\Omega}_{\varepsilon,\delta}(n^{1/6})$
Bottleneck	$\tilde{O}(\frac{\log n}{\varepsilon})$	$\tilde{O}(\frac{\sqrt{\log n \log 1/\delta}}{\varepsilon})$	N.A.

- Counting queries are harder to privatize

DP-ASRQ VS DP-APSD

	Graph	Privacy	Upper bound	Lower bound
DP-ASRQ (Counting query)	(Un)Weighted	Attribute	$\tilde{O}_{\varepsilon}(n^{1/3})$ $\tilde{O}_{\varepsilon,\delta}(n^{1/4})$	$\tilde{\Omega}_{\varepsilon,\delta}(n^{1/6})$
DP-APSD	Weighted	Edge weight (Stronger)	$\tilde{O}_{\varepsilon}(n^{2/3})$ $\tilde{O}_{\varepsilon}(n^{1/2})$	$\tilde{\Omega}_{\varepsilon,\delta}(n^{1/6})$

- Two problems share the same lower bound
- DP-APSD is a strictly harder problem than DP-ASRQ

Standard Notions and Tools

- Differential Privacy
    - Sensitivity
    - Laplace Mechanism
    - Gaussian Mechanism
    - Basic and Strong Composition Theorem
- Probability Theory
    - Sum of Laplace and Gaussian random variables
    - Concentration of Laplace and Gaussian random variables

ε-DP Algorithm for Counting Query

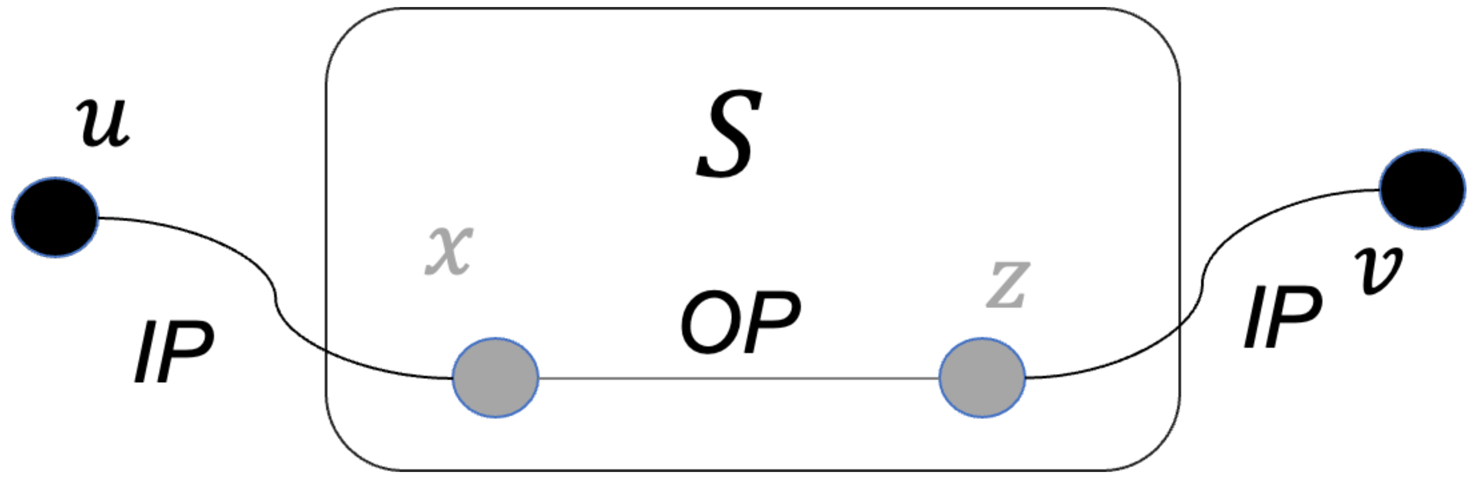
Two simple solutions:

- Input perturbation.
  - Add Laplace noise of Lap$(1/\varepsilon)$ to each attribute and return the counting.
  - If the path is long, then the error can be large.
- Output perturbation.
  - Compute the counting first, and add noise of Lap(n^2/ε) to the query output.
  - The sensitivity is n^2
- Both solutions lead to additive error of $\tilde{O}(n)$

Can we balance two regimes to reduce the error?

ε-DP Algorithm for Counting Query

Key idea: Carefully combine Input and Output perturbation



- Sample a set of shortcut vertices S, $|S| = s$
  - If the path is long, there will be vertices in S
- Decompose the path into paths in S and paths outside S
  - $f(u, v) = f(u, x) + f(x, z) + f(z, v)$
  - Apply Input perturbation on $f(u, x)$ and $f(z, v)$, output perturbation on $f(x, z)$

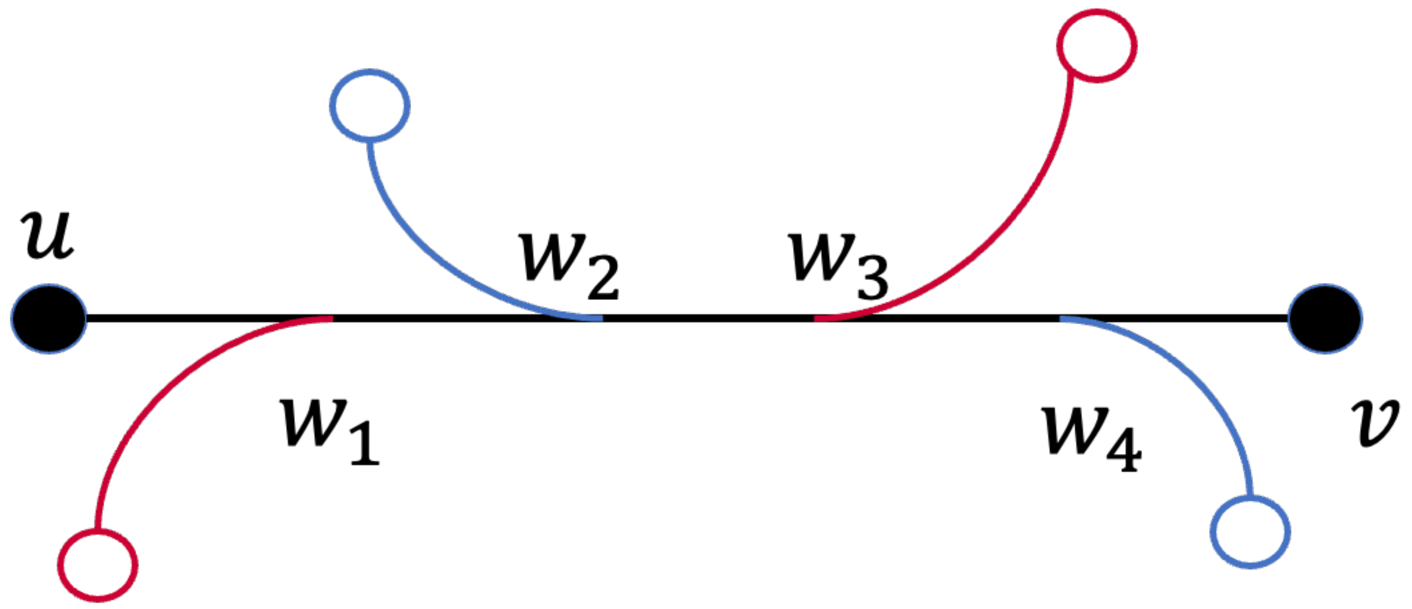
Towards our goal

- For input perturbation, shortcut set reduce the length of paths
- For output perturbation, we need to reduce the sensitivity

ε-DP Algorithm for Counting Query

Reduce the sensitivity of S: Canonical segments

- Canonical segments are sub-paths that no other shortest paths pass through



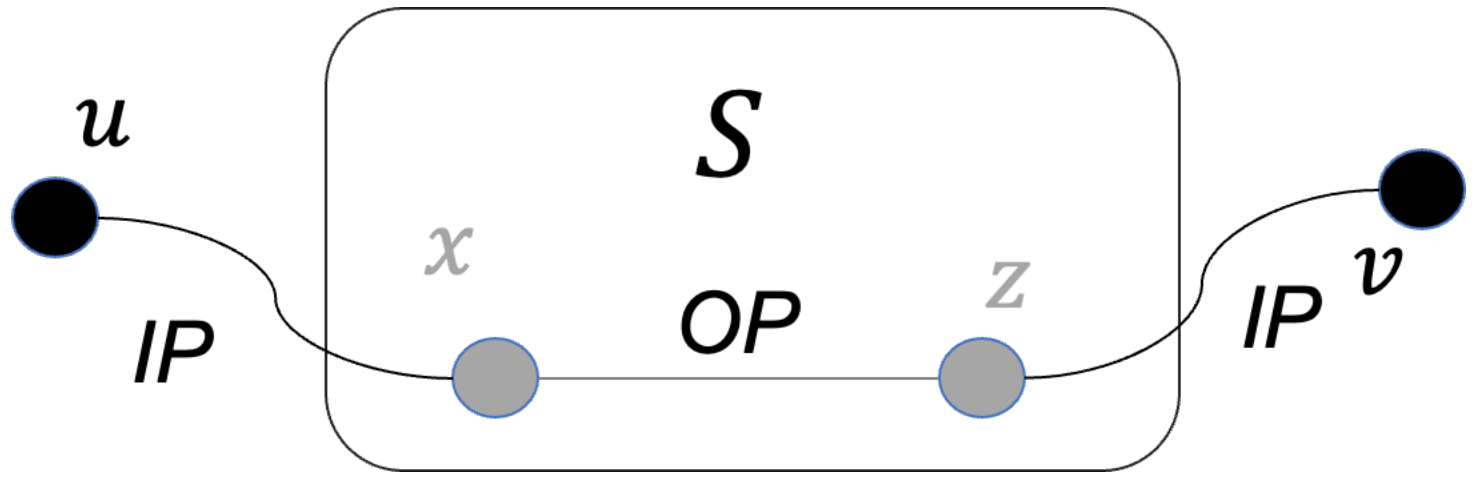
Consider Path(u, v) where $u, v \in S$, the canonical segments are:

$(u, w_1), (w_1, w_2), (w_2, w_3), (w_3, w_4), (w_4, v)$

- Properties of canonical segments
  - They are disjoint
  - Each canonical segment has sensitivity of 1
  - For shortcut set S, there are at most s^2 canonical segments, call the set Canon(S)

ε-DP Algorithm for Counting Query

Algorithm



- Sample shortcut set S, compute Canon(S).
- Add Independent Lap$(2/\varepsilon)$ to all edge attributes.
- Add Independent Lap$(2/\varepsilon)$ to each canonical segment attribute
- Report the counting query for $u, v \in V$:
  - If Path(u, v) does not have a vertex in S, return $\hat{f}(u, v)$
  - If Path(u, v) has one vertex z in S, return $\hat{f}(u, z) + \hat{f}(z, v)$
  - Else, take the first and last vertex x, z in S, return $\hat{f}(u, v) = \hat{f}(u, x) + \hat{f}(x, z) + \hat{f}(z, v)$

ε-DP Algorithm for Counting Query

Theorem 1

There exists an ε-differentially private algorithm for DP-ARSQ with additive error at most $\tilde{O}(n^{1/3}/\varepsilon)$ with high probability. That is, the algorithm outputs $\hat{f}$ such that

$$\Pr\Big(\max_{u,v \in V} |\hat{f}(u,v) - f(u,v)| = O(\frac{n^{1/3}\log^{5/6} n}{\varepsilon})\Big) \geq 1 - \frac{1}{n}$$

- Analysis sketch
  - Input perturbation (Step 2) has at most $\tilde{O}(n/s)$ additive error
  - Output perturbation (Step 3) has at most $\tilde{O}(s^2)$ additive error
  - Total error: $\tilde{O}(\sqrt{n/s + s^2})$ -- $s = n^{1/3}$ balances two terms

ε, δ-DP Algorithm for Counting Query

Key idea: Strong composition on single-source shortest path tree

(When the graph is a tree, the problem is a lot easier!)

Result (ε, δ-DP algorithm tree graphs)

There exists an (ε, δ)-differentially private algorithm for tree graphs with additive error at most $O(\log^{1.5} n\sqrt{\log 1/\delta}/\varepsilon)$ with high probability.

- Sample a set of shortcut vertices S
- Build single-source shortest path tree rooted at each vertex in S
- Privatize each tree and use strong composition.

An analog of output perturbation

ε, δ-DP Algorithm for Counting Query

Algorithm

- Sample shortcut set S, compute $T(v)$ for $v \in S$
- Run PrivateTree algorithm for each $T(v)$
- Apply strong composition on all private trees
- Add Gaussian noise of Gauss$(0,4/\varepsilon^2 \ln(2.5/\delta)\log n)$ to all edge attributes
- Report the counting query for $u, v \in V$:
  - If one of $u, v \in S$, return $\hat{f}_T(u, v)$
  - If $u, v \notin S$ but Path(u, v) has one vertex z in S, return $\hat{f}_T(u, z) + \hat{f}_T(z, v)$
  - Else, return $\hat{f}(u, v)$

ε, δ-DP Algorithm for Counting Query

Theorem 2

There exists an (ε, δ)-differentially private algorithm for DP-ARSQ with additive error at most $\tilde{O}(n^{1/4} \cdot \log^{1/2} 1/\delta/\varepsilon)$ with high probability. That is, the algorithm outputs $\hat{f}$ such that

$$\Pr\left(\max_{u,v \in V} |\hat{f}(u,v) - f(u,v)| = O(\frac{n^{1/4} \log^{1.25} n \sqrt{\log(1/\delta)}}{\varepsilon})\right) \geq 1 - \frac{1}{n}$$

- Analysis sketch
  - Input perturbation (Step 4) has at most $\tilde{O}(s)$ additive error
  - Private tree outputs (Step 3) have at most $\tilde{O}(n/s)$ additive error
  - Total error: $\tilde{O}(\sqrt{n/s + s})$ -- $s = n^{1/2}$ balances two terms

Private Algorithms for Bottleneck Query

Apply input perturbation suffices!

- For ε-DP, use Laplace mechanism
- For ε, δ-DP, use Gaussian mechanism

Result 4 (Private algorithms for bottleneck query)

There exists an ε-differentially private algorithm for DP-ARSQ with additive error at most $\tilde{O}(\log n/\varepsilon)$ with high probability; For (ε, δ)-DP the additive error is $\tilde{O}(\sqrt{\log n}\log(1/\delta)/\varepsilon)$

Open Problems

- Close the gap for both DP-APSD and DP-ASRQ
  - DP-APSD: $n^{1/6} \sim n^{1/2}$
  - DP-ASRQ: $n^{1/4} \sim n^{1/2}$
- Single-pair shortest distance and One-set range query

Thank you!