Cycle Counting Under Local Differential Privacy for Degeneracy-Bounded Graphs

Quentin Hillebrand
The University of Tokyo, Japan

Vorapong Suppakitpaisarn
The University of Tokyo, Japan

Tetsuo Shibuya
The University of Tokyo, Japan

Abstract

We propose an algorithm for counting the number of cycles under local differential privacy for degeneracy-bounded input graphs. Numerous studies have focused on counting the number of triangles under the privacy notion, demonstrating that the expected ℓ_2-error of these algorithms is $\Omega(n^{1.5})$, where n is the number of nodes in the graph. When parameterized by the number of cycles of length four (C_4), the best existing triangle counting algorithm has an error of $O(n^{1.5}+\sqrt{C_4}) = O(n^2)$. In this paper, we introduce an algorithm with an expected ℓ_2-error of $O(\delta^{1.5}n^{0.5} + \delta^{0.5}d_{\max}^{0.5}n^{0.5})$, where δ is the degeneracy and $d_{\max}$ is the maximum degree of the graph. For degeneracy-bounded graphs ($\delta \in \Theta(1)$) commonly found in practical social networks, our algorithm achieves an expected ℓ_2-error of $O(d_{\max}^{0.5}n^{0.5}) = O(n)$. Our algorithm's core idea is a precise count of triangles following a preprocessing step that approximately sorts the degree of all nodes. This approach can be extended to approximate the number of cycles of length k, maintaining a similar ℓ_2-error, namely $O(\delta^{(k-2)/2}d_{\max}^{0.5}n^{(k-2)/2} + \delta^{k/2}n^{(k-2)/2})$ or $O(d_{\max}^{0.5}n^{(k-2)/2}) = O(n^{(k-1)/2})$ for degeneracy-bounded graphs.

2012 ACM Subject Classification Security and privacy → Privacy-preserving protocols; Theory of computation → Graph algorithms analysis

Keywords and phrases Differential privacy, triangle counting, degeneracy, arboricity, graph theory, parameterized accuracy

Digital Object Identifier 10.4230/LIPIcs.STACS.2025.49

Funding *Quentin Hillebrand*: partially supported by KAKENHI Grant 20H05965, and by JST SPRING Grant Number JPMJSP2108.
Vorapong Suppakitpaisarn: partially supported by KAKENHI Grant 21H05845 and 23H04377.
Tetsuo Shibuya: partially supported by KAKENHI Grant 20H05967, 21H05052, and 23H03345.

Acknowledgements The authors wish to express their thanks to the anonymous reviewers whose valuable feedback greatly enhanced the quality of this paper.

1 Introduction

In recent years, differential privacy [13, 15] has become the gold standard for providing strong privacy guarantees while enabling meaningful data analysis. Differential privacy ensures that the output of a computation does not significantly change when any single individual's data is modified, thus safeguarding individual privacy. While much of the initial work in differential privacy focused on traditional tabular data [14, 26], there is increasing interest in extending these privacy guarantees to graph data [31, 35], which presents its own unique set of challenges.

42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025).
Editors: Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyễn Kim Thắng;
Article No. 49; pp. 49:1–49:22
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Differential privacy has evolved into numerous variants to accommodate different scenarios, as detailed in [8]. Of particular interest to us is the concept of local differential privacy [7, 17]. This variant is unique in that it does not rely on the assumption of a trusted central server. Instead, users must obfuscate their private data before sharing it with an untrusted computing entity. In the context of graph data, the most commonly adopted notion is edge local differential privacy [29], where the sensitive information of each user pertains to their connections with others.

A widely used obfuscation method is randomized response [33, 32]. In this approach, users invert each bit of their adjacency vector with a certain probability. The server then collects this distorted information to construct an obfuscated graph. Although it is possible to publish various graph statistics from the obfuscated graph, the resulting information tends to be imprecise. Algorithms specifically designed to publish particular statistics typically yield more accurate and useful graph information.

Table 1 Upper and lower bounds of the expected ℓ_2-error for triangle and k-cycle counting under the local differential privacy. "Interactive" refers to a scenario in which multiple rounds of communication between the server and the clients are permitted.

	Upper Bound	**Lower Bound**
Triangle	$O(n^2)$ ([22], general graphs) $O(d_{\max}^{1.5} n^{0.5})$ ([16], general graphs) $O(d_{\max}^{0.5} n^{0.5})$ (this work, degeneracy-bounded graph)	$\Omega(n^{1.5})$ (non-interactive) [24] $\Omega(n^{1.5})$ (interactive) [16] $\Omega(n^2)$ (non-interactive) [16]
Odd length cycles C_k	$O\left(n^{k-1}\right)$ (folklore, general graphs) $O\left(n^{(k-1)/2}\right)$ (this work, degeneracy-bounded graph)	

One graph statistic frequently considered by researchers in local differential privacy is the number of subgraphs [22, 20]. Specifically, many studies have focused on the publication of triangle counts [22, 23, 20, 16]. Theoretical analysis results on the ℓ_2-error are summarized in Table 1. Unfortunately, to date, when n is the number of nodes and $d_{\max}$ is the maximum degree of the input graphs, the best algorithm has an expected ℓ_2-error of $O(n^2)$ or $O(d_{\max}^{1.5} n^{0.5})$. We believe that this error is too large for many applications and should be improved. On the other hand, it has been shown that for all locally differentially private algorithms, there exists a class of graphs where the ℓ_2-error is $\Omega(n^{1.5})$ [16]. This lower bound implies that the expected ℓ_2-error cannot be significantly improved.

1.1 Our Contribution

This motivates us to consider a specific class of graphs. In this paper, we specifically focus on graphs with bounded degeneracy, as most social graphs exhibit degeneracy values that are substantially smaller than both the number of vertices and the maximum degree. The table in the Appendix of [12] provides statistics on a diverse range of graphs, detailing the number of nodes, degeneracy, and maximum degree. As shown in the paper, for sufficiently large graphs, degeneracy is consistently at least an order of magnitude smaller than the maximum degree, and in some instances, several orders of magnitude smaller.

Additionally, several synthetic graph models commonly considered realistic naturally produce graphs with low degeneracy. Examples include preferential attachment graphs [1] and bounded expansion graphs [27].

Degeneracy is particularly significant in parameterizing the complexity of subgraph counting algorithms, as demonstrated in [6, 2, 5]. Given the relationship between computational complexity and estimation error in triangle counting algorithms, degeneracy is an important parameter for characterizing accuracy.

Let the graph degeneracy be δ. We propose a locally differentially private algorithm with an expected ℓ_2-error of $O\left(\delta^{1.5}n^{0.5} + \delta^{0.5}d_{\max}^{0.5}n^{0.5}\right)$. When the graph degeneracy is bounded ($\delta = O(1)$), the expected ℓ_2-error becomes $O(d_{\max}^{0.5}n^{0.5}) = O(n)$. This result implies that our expected error for the degeneracy-bounded graphs can be smaller than the lower bound for general graphs.

We also extend our results to count the number of cycles with odd lengths in degeneracy-bounded graphs. To our knowledge, there are only two local differentially private algorithms proposed for counting subgraphs of more than three nodes. The first algorithm [24] is designed to count the number of four-length cycles but operates within the shuffle model, which is weaker than the original local differential privacy model. The second algorithm counts the number of walks of length k [3]. This field has limited work due to the significant noise introduced to ensure user privacy, which accumulates as the subgraph size increases. This accumulation results in unacceptable errors for differential privacy in larger subgraphs. For instance, while the expected ℓ_2-error from triangle counting algorithms based on randomized response is $O(n^2)$ [24], the expected ℓ_2-error for similar algorithms estimating the number of C_k is as high as $O(n^{k-1})$. In other words, the error increases by a factor of n with each increment in cycle length.

In this work, we propose an algorithm that significantly reduces the expected ℓ_2-error to $O(n^{(k-1)/2})$ in degeneracy-bounded graphs. We believe that this error is much smaller than the actual number of cycles in most graphs. Consequently, our algorithm is the first to publish a meaningful number of large cycles under local differential privacy.

1.2 Technical Overview

In this section, we provide an overview of the technical concepts behind our triangle counting algorithm. The algorithm for counting odd-length cycles, for $k \geq 5$, extends these ideas but requires a more intricate and detailed analysis.

Let the input graph be $G = (V = \{\nu_1, \ldots, \nu_n\}, E)$. In prior work [22], they apply a randomized response mechanism that flips each bit in the adjacency matrix with a certain probability. Let the resulting graph after applying the randomized response be $G' = (V, E')$. In the local differential privacy setting, each node ν_i knows whether it is connected to another node ν_j (where $\nu_j \neq \nu_i$) if $\{\nu_i, \nu_j\} \in E$. For the triangle counting method, node ν_i considers $(\nu_i, \nu_j, \nu_\kappa)$ as a triangle if $\{\nu_i, \nu_j\} \in E$, $\{\nu_i, \nu_\kappa\} \in E$, and $\{\nu_j, \nu_\kappa\} \in E'$. Define $e_{i,j,\kappa} = 1$ if node ν_i considers $(\nu_i, \nu_j, \nu_\kappa)$ as a triangle, otherwise set $e_{i,j,\kappa} = 0$. Define $S_i = \{(j,\kappa) : \{\nu_i, \nu_j\}, \{\nu_i, \nu_\kappa\} \in E \text{ and } j < \kappa\}$. The estimated number of triangles for node ν_i, reported by the user, is $\tilde{t}_i = \sum\limits_{(j,\kappa)\in S_i} e_{i,j,\kappa}$. The total estimated number of triangles in the graph is then $\tilde{f}_\Delta(G) = \frac{1}{3}\sum_i \tilde{t}_i = \frac{1}{3}\sum_i \sum_{(j,\kappa)\in S_i} e_{i,j,\kappa}$, where dividing by three corrects for the fact that each triangle is counted once by each of the three users forming it (i.e., triple-counted), ensuring each triangle is counted only once.

The ℓ_2-error of the estimated triangle count $\tilde{f}_\Delta(G)$ mostly arises from the variance in the estimation. A significant portion of this variance comes from the covariance between pairs of variables in the summation $\frac{1}{3}\sum\limits_i \sum\limits_{(j,\kappa)\in S_i} e_{i,j,\kappa}$. Two variables, $e_{i,j,\kappa}$ and $e_{i',j',\kappa'}$, are dependent if $(j,\kappa) = (j',\kappa')$. The number of dependent pairs in the counting process is equivalent to the number of tuples $(\nu_i, \nu_j, \nu_{i'}, \nu_\kappa)$ such that $(j,\kappa) \in S_i \cap S_{i'}$, which corresponds to the number of 4-cycles in the input graph G. Therefore, the squared ℓ_2-error is approximately proportional to the number of 4-cycles in the graph, which is $O(n^4)$.

Let us assume that the indices of all users are predetermined and publicly known before the counting process begins. Define $S_i' = \{(j, \kappa) : \{\nu_i, \nu_j\}, \{\nu_i, \nu_\kappa\} \in E \text{ and } j < i < \kappa\}$. If node i only considers the pairs (j, κ) within S_i', then each triangle is counted exactly once. The estimated number of triangles, $\hat{f}_\Delta(G)$, can be calculated as $\hat{f}_\Delta(G) = \sum_i \hat{t}_i$, where $\hat{t}_i = \sum\limits_{(j,\kappa)\in S_i'} e_{i,j,\kappa}$. In this counting method, the number of dependent variable pairs is at most the number of 4-cycles that contain the three nodes ν_i, ν_j, ν_κ with $j < i < \kappa$.

Let δ represent the degeneracy of the input graph G, and for each $\nu \in V$, let $d(\nu)$ denote the degree of ν. Assume that the degrees of all nodes are publicly known, and the nodes are indexed in non-decreasing order of their degree, i.e., if $i > j$, then $d(\nu_i) \leq d(\nu_j)$. Referring to the bound established by Chiba and Nishizeki [6], which states that $\sum\limits_{(\nu_i,\nu_j)\in E} \min(d_i, d_j) \leq O(\delta \cdot |E|)$, we demonstrate in this paper that the number of such cycles is $O(\delta^3 n)$. Consequently, the squared ℓ_2-error is reduced from $O(n^4)$ in previous work to $O(\delta^3 n)$.

However, we cannot assume that the degrees of all nodes are publicly known, as this information is sensitive. To address this issue, we use local Laplacian queries, allowing each user to publish a noisy version of their degree. Let the noisy degree of $\nu \in V$ be denoted as $\tilde{d}(\nu)$. We then assign indices to users based on these noisy degrees, such that if $i > j$, then $\tilde{d}(\nu_i) \leq \tilde{d}(\nu_j)$. Afterward, we run the protocol described in the previous paragraph. We show that even with noisy degrees, the expected number of such cycles remains bounded by $O(\delta^3 n)$.

In summary, our mechanism involves two steps. First, users publish their noisy degrees using the local Laplacian mechanism, and the server assigns indexes based on these noisy values. In the second step, using the results of randomized response, each user ν_i estimates the number of triangles $(\nu_i, \nu_j, \nu_\kappa)$ where $j < i < \kappa$. This method significantly reduces the number of dependent triangle pairs in degeneracy-bounded graphs, which in turn lowers the variance of the estimation.

1.3 Related Works

The field of graph data mining under local differential privacy is relatively new. In contrast, differential privacy has been studied for many years by various researchers, including works like [18, 28]. According to [22], local differential privacy typically only hides edges or relationships, except in special cases like [36]. Differential privacy, on the other hand, can hide whether an individual or node is part of a social network, as shown in [19, 30]. Therefore, while both edge and node differential privacy exist, node differential privacy does not apply in the context of local differential privacy.

Recent works have proposed methods to estimate the densest subgraph, k-core decomposition, and degeneracy under local differential privacy [10, 9, 11]. However, since we are focused on estimating different graph statistics in graphs, we do not use or extend the ideas from these works. Instead, the estimation of degeneracy can be used to approximate the ℓ_2-error of our algorithm.

2 Preliminaries

2.1 Notations

For $V = \{\nu_1, \ldots, \nu_n\}$ a set of vertices and $E \subseteq V^2$ a set of edges, we denote by $G = (V, E)$ the graph on V. We consider simple undirected graphs, meaning that for $\nu, \nu' \in V$, $(\nu, \nu) \notin E$ and $(\nu, \nu') \in E \implies (\nu', \nu) \in E$. We denote by $n = |V|$ the size of the graph and $m = |E|$ its number of edges.

For each $i \in [1, n]$, we introduce $a_i = [a_{i,1}, \dots, a_{i,n}]$, the adjacency list of user ν_i, where for any $j \in [1, n]$, $a_{i,j} = 1$ if the edge (ν_i, ν_j) is in E and $a_{i,j} = 0$ otherwise. Additionally, we introduce d_i, the degree of node ν_i, which corresponds to the number of edges incident to ν_i.

We call a path of length $k \in \mathbb{N}$, denoted P_k, any tuple $(\nu_{l_1}, \dots, \nu_{l_k})$ such that, for all $i \in [1, k]$, $(\nu_{l_i}, \nu_{l_{i+1}}) \in E$, and, for all $i \neq j$, $\nu_{l_i} \neq \nu_{l_j}$. We also use $\#P_k(G)$ to refer to the number of paths of length k in G. Similarly, a cycle of length $k \in \mathbb{N}$, or C_k, is a tuple $(\nu_{l_1}, \dots, \nu_{l_k})$ that forms a path and satisfies $(\nu_{l_k}, \nu_{l_1}) \in E$. We will also use $\#C_k(G)$ to refer to the number of cycles of length k in G.

2.2 Edge Local Differential Privacy

We say that two adjacency lists a and a' are neighboring if they differ by one bit, i.e. if we can go from one to the other by adding or removing an edge to node ν_i. If a' is a neighbor of a, we write that $a \sim a'$. The notion of edge local differential privacy is as follows:

▶ **Definition 1** (ε-edge local differentially private query). *Let $\varepsilon > 0$. A randomized algorithm $\mathcal{R}$ is a ε-edge local differentially private query on the node ν_i if, for all neighboring bit strings $a \sim a'$, and for all S, it holds that*

$$\mathbb{P}\left[\mathcal{R}(a) \in S\right] \leq e^{\varepsilon} \mathbb{P}\left[\mathcal{R}(a') \in S\right].$$

▶ **Definition 2** (ε-edge local differentially private algorithm [29]). *Let $\mathcal{A}$ be an algorithm that generates multiple randomized queries for each user, has each user apply these queries to their adjacency vector, and then estimates some graph statistics based on the results. We say $\mathcal{A}$ is an ε-edge local differentially private algorithm if, for all users ν_i and for all possible sets of queries $\mathcal{R}_1, \dots, \mathcal{R}_k$ inquired to ν_i (where for each $1 \leq j \leq k$, $\mathcal{R}_j$ is an ε_j-edge local differentially private query), it holds that $\varepsilon_1 + \dots + \varepsilon_k \leq \varepsilon$.*

2.3 Laplacian Query and Restricted Sensitivity

Next, we introduce queries that are ε-edge local differentially private. We first consider a query which aims to give an estimate of a real number statistics of the adjacency vector.

▶ **Definition 3** (Edge local Laplacian query [21]). *For a function $f : \{0,1\}^n \to \mathbb{R}$ on adjacency lists, and $a \sim a'$ denoting neighboring adjacency lists, the global sensitivity of f is defined as $\Delta_f = \max_{a \sim a'} |f(a) - f(a')|$. For $\varepsilon > 0$, the query that outputs $f(a) + \mathtt{Lap}(\Delta_f/\varepsilon)$ is ε-edge local differentially private, where $\mathtt{Lap}(b)$ represents noise drawn from the Laplacian distribution with parameter b.*

Global sensitivity in Definition 3 is designed to handle the worst-case scenario, which can lead to large amounts of noise being added to the data when using the Laplacian mechanism. However, if the data is known to belong to a specific set, restricted sensitivity allows us to adjust the noise according to the sensitivity within that set, resulting in more tailored and potentially lower noise levels.

▶ **Definition 4** (Restricted sensitivity (Definition 8 of [4])). *Let $a = (a_1, \dots, a_n), a' = (a'_1, \dots, a'_n) \in \{0,1\}^n$ and $d(a, a')$ be the Hamming distance between a and a'. The restricted sensitivity of f over a set of possible output $\mathcal{H}$ is*

$$RS_f(\mathcal{H}) = \max_{a,a' \in \mathcal{H}} \left(\frac{|f(a) - f(a')|}{d(a, a')} \right).$$

We can use restricted sensitivity to publish data even if it is not initially in the set. To do this, we first need to define a projection method to map the data to the set. In this work, we will consider $\mathcal{H}_d$, the class of adjacency list with a maximum degree of d, for calculating restricted sensitivity. We assume that the order of all nodes is fixed, and if a node ν_i is adjacent to more than d nodes, we retain only the first d nodes according to this order. The map can be considered as an operation on each adjacency vector a_i. We denote the mapping result on a_i as $\mu_d(a_i)$.

▶ **Definition 5** (Edge local Laplacian query with restricted sensitivity on $\mathcal{H}_d$ [4]). *For any f queried to a user i, the query that answers $f(\mu(a_i)) + \mathsf{Lap}(3 \cdot RS_f(\mathcal{H}_d)/\varepsilon)$ is called edge local Laplacian query with restricted sensitivity on $\mathcal{H}_d$, and provides ε-edge local differential privacy.*

2.4 Unbiased Randomized Response

In this subsection, we consider the randomized response query, which aims to publish an obfuscated adjacency vector.

▶ **Definition 6** (Randomized response query [33, 32]). *For $\varepsilon > 0$, the randomized response mechanism takes an adjacency list $a = (a_1, \dots, a_n)$ as input and outputs an obfuscated list $\tilde{a} = (\tilde{a}_1, \dots, \tilde{a}_n)$. For i, the probability that $\tilde{a}_i$ is set to 1 is given by:*

$$\mathbb{P}\left[\mathtt{RR}(\tilde{a}_i) = 1\right] = \begin{cases} \frac{e^\varepsilon}{1+e^\varepsilon} & \textit{if } a_i = 1 \\ \frac{1}{1+e^\varepsilon} & \textit{if } a_i = 0. \end{cases}$$

With this definition, randomized response provides ε-edge local differential privacy.

We can construct a graph $\tilde{G}$ based on the collection of obfuscated adjacency vectors obtained from all users. Using the statistics of the obfuscated graph $\tilde{G}$, we can then publish various information, including the number of subgraphs [34, 22, 23, 20]. However, randomized response produces biased results, making it less suitable for counting queries. This bias can be fixed by the subsequent definition.

▶ **Definition 7** (Unbiased randomized response query [16]). *Let $\varepsilon > 0$ and $\tilde{a}_i$ be the adjacency vector published through randomized response with budget ε by user ν_i. Then, for all $(i, j) \in [1, n]^2$,*

$$\hat{a}_{i,j} = \frac{e^\varepsilon + 1}{e^\varepsilon - 1}\tilde{a}_{i,j} - \frac{1}{e^\varepsilon - 1}$$

is an unbiased estimator of $a_{i,j}$. Additionally, for $(i, j) \neq (i', j')$, $\hat{a}_{i,j}$ is independent of $\hat{a}_{i',j'}$, and $Var(\hat{a}_{i,j}) = \frac{e^\varepsilon}{(e^\varepsilon-1)^2}$. We refer to a query that publishes $\hat{a}_i$ as the unbiased randomized response query. This query is ε-edge locally differentially private.

We can use the results from the unbiased randomized response query to calculate the number of subgraphs. For example, without privacy constraints, the number of triangles can be calculated as $\sum\limits_{i<j<k} a_{i,j} \cdot a_{j,k} \cdot a_{k,i}$. To privately estimate the number of triangles, we use $\sum\limits_{i<j<k} \hat{a}_{i,j} \cdot \hat{a}_{j,k} \cdot \hat{a}_{k,i}$. It is theoretically shown in [16] that the estimator $\sum\limits_{i<j<k} \hat{a}_{i,j} \cdot \hat{a}_{j,k} \cdot \hat{a}_{k,i}$ has a smaller ℓ_2-error compared to the estimator obtained from the randomized response query, $\sum\limits_{i<j<k} \tilde{a}_{i,j} \cdot \tilde{a}_{j,k} \cdot \tilde{a}_{k,i}$.

2.5 Graph Arboricity and Degeneracy

Graph arboricity and degeneracy can be defined as follows:

▶ **Definition 8** (Arboricity). *The arboricity of a graph G is the minimal number $\alpha(G)$ such that the edges of G can be partitioned into $\alpha(G)$ forests.*

▶ **Definition 9** (Degeneracy). *The degeneracy of a graph G is the smallest number $\delta(G)$ such that any subgraph of G, contains at least one node with induced degree at most $\delta(G)$.*

We observe that the variable δ is frequently used as a privacy parameter in differential privacy. However, since we do not consider that parameter in this paper, we choose to use δ to represent degeneracy, which is also a common convention. When the context is clear, we will drop the G of the notation and simply write α and δ. The two quantities are linked by the following theorem.

▶ **Theorem 10** (equation 3 and lemma 2.2 of [37]). *In any graph G, degeneracy and arboricity satisfy $\alpha \leq \delta \leq 2\alpha - 1$.*

The arboricity has previously been used outside of the differential private community to bound some graph statistics. A folklore useful result is that the number of edges in a graph is smaller than δn. Another well known result is as follows:

▶ **Theorem 11** (Chiba-Nishizeki Bound [6]). *With $m = |E|$ and d_i the degree of node ν_i, then*

$$\sum_{(\nu_i, \nu_j) \in E} \min(d_i, d_j) \leq m\alpha.$$

3 Node-Reordered Graphs and Their Properties

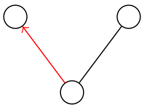
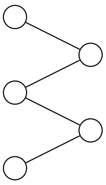
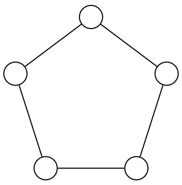
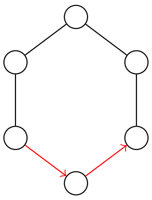
The first step of our mechanism is to order the vertices based on their estimated degree. The algorithm for this step is shown in Algorithm 1. At Line 2 of the algorithm, we privately publish the estimated degree. Under edge local differential privacy, the global sensitivity of the degree is 1. Therefore, we can use the Laplacian query (Definition 3) with noise scaled to $1/\varepsilon_0$ to publish the degree, where ε_0 is the privacy budget allocated to this step. We denote the estimated degree as $\tilde{d}_i = d_i + \mathrm{Lap}(1/\varepsilon_0)$.

Algorithm 1 Calculate a low degree ordering of a graph with respect to the estimated degree.

```
1 Function GetOrdering
      Input: Graph $G = (V, E)$, privacy budget $\varepsilon_0$
      Output: A low degree ordering $\phi$ of $G$ with respect to the estimated degree
2     [User $i$] Calculate and send $\tilde{d}_i \leftarrow d_i + \mathtt{Lap}(\frac{1}{\varepsilon_0})$ to the central server
3     [Server] Let $\phi(i) = j$ if $\tilde{d}_i$ is the $j$-the largest number in $\tilde{d}_1, \ldots, \tilde{d}_n$. Calculate
        $\phi(i)$ for all $i$
4     return $\phi$;
```

After publishing the estimated degrees, in Line 3, we assign an order ϕ to the nodes based on their degrees, which we refer to as a *low degree ordering*. For $G = (\{\nu_1, \ldots, \nu_n\}, E)$, we denote the reordered graph as $G^\phi = (V^\phi, E^\phi)$, where $V^\phi = \{\eta_i \mid i \in [1, n]\}$ and $\nu_i = \eta_{\phi(i)}$ for all i. The edge set E^ϕ is defined as $\{(\eta_{\phi(i)}, \eta_{\phi(j)}) \mid (\nu_i, \nu_j) \in E\}$. We note that G and G^ϕ are isomorphic, and thus have the same number of subgraphs. We denote by $d_i(G^\phi)$ the degree of η_i in G^ϕ and $d_i^-(G^\phi)$ the number of neighbors of node η_i in the set $\{\eta_1, \ldots, \eta_{i-1}\}$.

Table 2 List of subgraphs analyzed in Section 3, including their representations and bounds on their counts in the graph produced by Algorithm 1. Oriented edges indicate directionality, with an arrow from ν_j to ν_i signifying that $j > i$. The bound on S_2^* aligns with the bound on S_2 presented in [6], but it constitutes a distinct contribution as it is established for imperfectly ordered graphs. In contrast, the result on C_{2k}^* is entirely novel to this work and serves as the primary result of our proof.

Symbol	S_2^*	P_k	C_k	C_{2k}^*
Representation				
Bound	$\mathcal{O}\left(\delta^2 n\right)$	$\mathcal{O}\left(\delta^{\lceil \frac{k}{2} \rceil} n^{\lfloor \frac{k}{2} \rfloor + 1}\right)$	$\mathcal{O}\left(\delta^{\lceil \frac{k}{2} \rceil} n^{\lfloor \frac{k}{2} \rfloor}\right)$ [25]	$\mathcal{O}\left(\delta^{k+1} n^{k-1}\right)$

In the remainder of this section, we analyze the properties of graphs produced by the reordering. Specifically, our focus is on bounding the frequency of certain substructures within the reordered graph. A summary of the results from this section is provided in Table 2.

▶ **Definition 12** (low star). *For $k \in \mathbb{N}^*$, a low-k-star is a subgraph consisting of a central node and k neighboring nodes, where at least one of the neighboring nodes has an index smaller than that of the central node. We denote by $S_k^*(G)$ the number of such subgraphs contained in a graph G.*

▶ **Theorem 13.** $\mathbb{E}\left[S_2^*(G^\phi)\right] \leq \mathcal{O}\left(\delta^2 n\right)$.

Proof. Let $\mathcal{N}_i(G^\phi)$ be the set of neighbors of η_i in G^ϕ. We have that:

$$S_2^*(G^\phi) = \sum_{i=1}^{n} d_i^-(G^\phi)(d_i(G^\phi) - 1) \leq \sum_{i=1}^{n} d_i(G^\phi) \times d_i^-(G^\phi) = \sum_{i=1}^{n} d_i(G^\phi) \sum_{\eta_j \in \mathcal{N}_i(G^\phi)} \mathbb{1}_{j<i}$$

$$= \sum_{(\eta_i, \eta_j) \in E^\phi} d_{\max(i,j)}(G^\phi)$$

Let τ_i denote the noise added to the estimated degree of user i. For each edge (η_i, η_j), their ranks can only be exchanged if the sum of the errors in both degree estimations exceeds the gap between the two degrees. Therefore, the quantity $d_{\max(i,j)}(G^\phi)$ satisfies

$$d_{\max(i,j)}(G^\phi) \leq \min(d_i, d_j) + |\tau_i| + |\tau_j|.$$

Using this inequality, we can rewrite the count of $S_2^*(G^\phi)$ as

$$S_2^*(G^\phi) \leq \sum_{(\eta_i, \eta_j) \in E^\phi} \min\left(d_i, d_j\right) + \sum_{i=1}^{n} |\tau_i| d_i.$$

Since τ_i is sampled from $\texttt{Lap}(1/\varepsilon_0)$, we have that $|\tau_i|$ follows an exponential law of expectation $1/\varepsilon_0$. Hence,

$$\mathbb{E}\left[S_2^*(G^\phi)\right] \leq \sum_{(\nu_i, \nu_j) \in E^\phi} \min\left(d_i, d_j\right) + \frac{m}{\varepsilon_0}.$$

Since G is isomorphic to G^ϕ, $\alpha(G) = \alpha(G^\phi)$ and using Theorem 11 it follows that

$$\sum_{(\nu_i,\nu_j)\in E^\phi} \min\left(d_i, d_j\right) \leq m \cdot \alpha(G).$$

Since $m \leq n\delta$ and $\alpha(G) = O(\delta)$, this gives $\mathbb{E}\left[S_2^*(G^\phi)\right] \leq \mathcal{O}\left(\delta^2 n\right)$. ◀

In addition to the ordered stars we just discussed, arboricity can also be used to bound the number of paths and cycles in a graph, as demonstrated in the following lemma and theorem. Recall that $\#P_k(G)$ is the number of paths with length k in the graph G.

▶ **Lemma 14.** *For any positive integer k, $\#P_{2k}(G) = \mathcal{O}\left(\delta^k n^{k+1}\right), \#P_{2k+1} = \mathcal{O}\left(\delta^{k+1} n^{k+1}\right)$.*

Proof. We first consider $\#P_{2k+1}(G)$. Let f be a function that maps a path of length $2k+1$ to a tuple of $k+1$ edges, defined as $f(e_1, \ldots, e_{2k+1}) = (e_1, e_3, \ldots, e_{2k+1})$. We observe that, for any tuple of $k+1$ edges denoted by $\mathcal{E} = (e'_1, \ldots, e'_{k+1})$, $f^{-1}(\mathcal{E})$ is either a set containing one path or an empty set. There is at most one path that uses e'_i as the $(2i-1)$-th edge of the path for all i. Thus, we can conclude that the number of paths of length $2k+1$ is at most the number of sets of $k+1$ edges, which is $m^{k+1} = \mathcal{O}\left(\delta^{k+1} n^{k+1}\right)$.

Next, let us consider $\#P_{2k}(G)$. Let f be a function that maps a path of length $2k$ to a tuple of k edges, defined as $f(e_1, \ldots, e_{2k}) = (e_1, e_3, \ldots, e_{2k-1})$. We observe that, for any tuple of k edges denoted by $\mathcal{E} = (e'_1, \ldots, e'_k)$, $f^{-1}(\mathcal{E})$ is a set of size no larger than n. There is at most one path of length $2k-1$ that uses e'_i as the $(2i-1)$-th edge of the path, and there are at most n possible ways to extend a path of length $2k-1$ to a path of length k. Hence, $\#P_{2k}(G) \leq n \cdot m^k = \mathcal{O}\left(\delta^k n^{k+1}\right)$. ◀

Recall that $\#C_k(G)$ is the number of cycles with size k in the graph G. We obtain the following theorem.

▶ **Theorem 15.** *For any $k \geq 1$, $\#C_{k+2}(G) \leq \frac{2}{k}\alpha(G)\#P_k(G)$.*

Proof. Let us denote $\#P_k^{(i)}$ the number of paths of length k that have node ν_i as an extremity and $\#C_k^{(i,j)}$ the number of cycles of length k containing edge (ν_i, ν_j). Using these notations, we have $\#C_{k+2} = \frac{1}{k} \sum_{(\nu_i,\nu_j)\in E} \#C_{k+2}^{(i,j)}$. Consider the number $\#C_{k+2}^{(i,j)}$. For a path of length k that has a node ν_i as a terminal, there is at most one cycle of length $k+2$ which includes this path and the edge (ν_i, ν_j). Therefore, we conclude that $\#C_{k+2}^{(i,j)} \leq \#P_k^{(i)}$. Similarly, we have $\#C_{k+2}^{(i,j)} \leq \#P_k^{(j)}$. Hence,

$$\#C_{k+2} \leq \frac{1}{k} \sum_{(\nu_i,\nu_j)\in E} \min\left(\#P_k^{(i)}, \#P_k^{(j)}\right).$$

For any function $h : E \to \{1, \ldots, n\}$ such that for all $e = (\nu_i, \nu_j) \in E$, $h(e)$ is equal to either i or j, $\min\left(\#P_k^{(i)}, \#P_k^{(j)}\right) \leq \#P_k^{(h(\nu_i,\nu_j))}$. By definition of the arboricity, there exists a set of disjoint forests $\{F_l\}_{l=1,\ldots,\alpha(G)}$ such that $E = \bigcup_{l=1}^{\alpha(G)} F_l$. By choosing a root for each tree of these forests, we can introduce a function h such that each edge has its child node as an image. In this way, each node can only be the image of one edge per forest. This leads to

$$\begin{aligned}\#C_{k+2} &\leq \frac{1}{k}\sum_{l=1}^{\alpha(G)} \sum_{(\nu_i,\nu_j)\in F_l} \min\left(\#P_k^{(i)}, \#P_k^{(j)}\right) \leq \frac{1}{k}\sum_{l=1}^{\alpha(G)} \sum_{e\in F_l} \#P_k^{(h(e))} \leq \frac{1}{k}\sum_{l=1}^{\alpha(G)} \sum_{i\in V} \#P_k^{(i)} \\ &= \frac{2}{k}\alpha(G)\#P_k.\end{aligned}$$

The last step is justified by the fact that each path having two extremities, the sum of all the paths of length k starting with node ν_i is twice the number of paths of length k. ◀

Combining Lemma 14 and Theorem 15, we obtain the following corollary[1].

▶ **Corollary 16.** *For $k \geq 1$, $\#C_{2k+2} = \mathcal{O}\left(\delta^{k+1}n^{k+1}\right)$ and $\#C_{2k+1} = \mathcal{O}\left(\delta^{k+1}n^{k}\right)$.*

Next, we focus on the number of cycles of length $2k$ for any $k \geq 2$, in which three consecutive vertices of the cycle exhibit monotonic ranks C^*_{2k}, as illustrated in Table 2. Throughout the rest of this article, we will denote the count of such subgraphs in G by $\#C^*_{2k}(G)$, omitting G from the notation when the context is clear. In the following theorem, for simplicity, we extend the notation by assuming $\#P_{-1}(G) = 1$ and $\#P_0(G) = n$ for every graph G.

▶ **Theorem 17.** *For $k \geq 2$, $\#C^*_{2k}(G) \leq 2\alpha(G)S^*_2(G)\#P_{2k-5}(G)$.*

Proof. Let $\#C^{*(i,j)}_{2k}(G)$ represent the number of subgraphs in G where three consecutive vertices exhibit monotonic ranks, with (ν_i, ν_j) being the edge immediately following these consecutive vertices. Also, for $k \geq 2$, let the number of paths of length p with a low-2-star as one of its extremities be denoted as $\#P^*_p$. Since we can construct at most one path included in $\#P^*_p$ where a low-2-star and a path of length $p-3$ are its extremities, we obtain the inequality $\#P^*_p \leq S^*_2 \cdot \#P_{p-3}$.

Let $C^{*(i,j)}_{2k}$ be a cycle which is counted in $\#C^{*(i,j)}_{2k}$. Consider the path in $C^{*(i,j)}_{2k}$ of length $2k-2$ starting from ν_i that does not pass through ν_j and the other path in $C^{*(i,j)}_{2k}$ of the same length starting from ν_j that does not pass through ν_i. We observe that one extremity of the two paths is a low-2-star. Hence, $\#C^{*(i,j)}_{2k} \leq \min\left(\#P^{*(i)}_{2k-2}, \#P^{*(j)}_{2k-2}\right)$ when $\#P^{*(i)}_p$ is the number of paths in the count of $\#P^*_p$ that have ν_i as an extremity. Using the same definition of h as in the proof of Theorem 15, we have

$$\begin{aligned}
\#C^*_{2k}(G) &\leq \sum_{(\nu_i,\nu_j)\in E} \#C^{*(i,j)}_{2k} \leq \sum_{(\nu_i,\nu_j)\in E} \min\left(\#P^{*(i)}_{2k-2}, \#P^{*(j)}_{2k-2}\right)\\
&\leq \sum_{l=1}^{\alpha(G)} \sum_{(\nu_i,\nu_j)\in F_l} \min\left(\#P^{*(i)}_{2k-2}, \#P^{*(j)}_{2k-2}\right) \leq \sum_{l=1}^{\alpha(G)} \sum_{e\in F_l} \#P^{*(h(e))}_{2k-2}\\
&\leq \sum_{l=1}^{a(G)} \sum_{i\in V} \#P^{*(i)}_{2k-2} \leq 2\alpha(G)\#P^*_{2k-2} \leq 2\alpha(G)S^*_2\#P_{2k-5}.
\end{aligned}$$ ◀

The next corollary follows Theorem 13, 17, and Lemma 14.

▶ **Corollary 18.** *For $k \geq 2$, $\mathbb{E}[C^*_{2k}(G^\phi)] = \mathcal{O}\left(\delta^{k+1}n^{k-1}\right)$.*

The next corollary considers the number of edge sets in G^ϕ with specific properties.

▶ **Corollary 19.** *For any $p \in \mathbb{N}$, we consider edge sets $\mathsf{E} \subseteq E^\phi$ of size $2p$ such that 1) for some $c > 0$, there exists a set of cycles $C_1, \ldots, C_c$ in G^ϕ where $C_1 \cup \cdots \cup C_c = \mathsf{E}$ and $C_i \cap C_j = \emptyset$ for $i \neq j$, and 2) at least one of $C_1, \ldots, C_c$ contains three consecutive vertices of monotonic index. The number of such edge sets is $\mathcal{O}\left(\delta^{p+1}n^{p-1}\right)$.*

[1] We note that this result was independently established in [25] by a different proof. We are grateful to the anonymous reviewer for bringing this to our attention.

Proof. Consider a partition of $2p$, denoted by $(p_1, \ldots, p_c)$, where $p_1 + \cdots + p_c = 2p$. The number of such partitions is a function of p and can be considered constant. We will demonstrate that the number of cycle sets $C_1, \ldots, C_c$ satisfying the conditions in the corollary statement, with $|C_i| = p_i$, is at most $\mathcal{O}\left(\delta^{p+1} n^{p-1}\right)$. Therefore, the number of cycle sets satisfying the corollary statement is no more than $\mathcal{O}\left(\delta^{p+1} n^{p-1}\right)$.

To prove the bound, we will consider two cases: either all the cycles have even lengths, or at least two of them have odd lengths, given that the total number of edges is even.

If all the cycles are of even length, then, for some $q > 0$ one of them is of length $2q$ and includes 3 consecutive vertices of monotonic index. By Corollary 18, there are $\mathcal{O}\left(\delta^{q+1} n^{q-1}\right)$ possibilities for this cycle. For the remaining cycles, Corollary 16 tells us that the number of admissible configurations is bounded by $\mathcal{O}\left(\delta^{p-q} n^{p-q}\right)$. In total, this gives a $\mathcal{O}\left(\delta^{p+1} n^{p-1}\right)$ bound. If at least two cycles have odd lengths, say $2q+1$ and $2r+1$, then by Corollary 16, the number of possible configurations for these cycles can be bounded by $\mathcal{O}\left(\delta^{q+1} n^{q}\right)$ for the first cycle and $\mathcal{O}\left(\delta^{r+1} n^{r}\right)$ for the second cycle, and $\mathcal{O}\left(\delta^{p-q-r-1} n^{p-q-r-1}\right)$ for the remaining cycles. Overall, this results in a bound of $\mathcal{O}\left(\delta^{p+1} n^{p-1}\right)$. ◀

4 Triangle Counting Algorithm

We propose Algorithm 2 to count the number of triangles based on the ordering and properties discussed in the previous section. First, we execute Algorithm 1 at Line 2. Next, at Line 3, we use the randomized response query to obtain an obfuscated graph. From Lines 4 to 8, we employ the Laplacian query with restricted sensitivity on $\mathcal{H}_d$ (Definition 5) to estimate the number of triangles associated with User i. Finally, at Line 9, we sum all the estimates and report the total as the estimated triangle count. We adopt the concept from [22] of distributing randomized response results to all nodes and having each node estimate its number of triangles. However, the other algorithmic ideas presented in this work are novel. In the following theorem, we demonstrate that our algorithm is differentially private.

Algorithm 2 Our algorithm for estimating the number of triangles in degeneracy-bounded graphs.

```
1  Function TriangleCounting
       Input: Graph $G = (V, E)$, privacy budget $\varepsilon = \varepsilon_0 + \varepsilon_1 + \varepsilon_2$, parameter $\zeta$
       Output: Estimation of the number of triangles in $G$
2      [All Users and Server] $\phi \leftarrow$ GetOrdering$(G, \varepsilon_0)$ (Algorithm 1);
3      [All Users and Server] Inquire the unbiased randomized response query with privacy
        budget $\varepsilon_1$ to all users. Let $(\hat{a}^{\phi}_{j,k})$ represent the results collected from this query. The
        server then distributes $(\hat{a}^{\phi}_{j,k})$ to all users.
4      [User $i$] $\hat{d}^{\phi}_i \leftarrow \tilde{d}^{\phi}_i + \frac{1}{\varepsilon_0}\ln(n/\zeta)$;
5      [User $i$] $a^{\phi}_i \leftarrow \mu_{\hat{d}^{\phi}_i}(a^{\phi}_i)$ (The function $\mu_d$ is defined before Definition 5.) ;
6      [User $i$] $S_i \leftarrow \{(j,k) \mid a^{\phi}_{i,j} = a^{\phi}_{i,k} = 1, j < i < k\}$;
7      [User $i$] $\hat{t}_i \leftarrow \sum_{(j,k)\in S_i} \hat{a}^{\phi}_{j,k}$;
8      [User $i$] $\tilde{t}_i \leftarrow \hat{t}_i + 3 \cdot \mathsf{Lap}(\frac{e^{\varepsilon_1}+1}{e^{\varepsilon_1}-1} \cdot \frac{\hat{d}^{\phi}_i}{\varepsilon_2})$;
9      [User $i$] Upload $\tilde{t}_i$ to the central server;
10     [Server] $\hat{f}_{\triangle}(G) \leftarrow \sum_{\nu_i \in V} \tilde{t}_i$;
11     return $\hat{f}_{\triangle}(G)$;
```

▶ **Theorem 20.** *Algorithm 2 provides $(\varepsilon_0 + \varepsilon_1 + \varepsilon_2)$-edge local differential privacy.*

Proof. For all possible executions of Algorithm 2, it inquires three queries to all users. They are 1) the Laplacian query with privacy budget ε_0 inside the `GetOrdering` function at Line 2, 2) the unbiased randomized response query with privacy budget ε_1 at Line 3, and 3) the Laplacian query with restricted sensitivity on $\mathcal{H}_d$ at Lines 4-8.

To prove this theorem, we only need to show that the query at Lines 4-8 is ε_2-edge local differentially private. The query aims to publish $f(a_i^\phi) = \sum_{(j,k)\in S_i} \hat{a}_{j,k}^\phi$. By the unbiased randomized response in Line 3, we have that, for any j, k, j', k', $|\hat{a}_{j,k}^\phi - \hat{a}_{j',k'}^\phi| \leq \frac{e^{\varepsilon_1}+1}{e^{\varepsilon_1}-1}$. It can be shown that, for $a_i^\phi, a_i^{'\phi} \in \mathcal{H}_{\hat{d}_i^\phi}$ (defined in Definition 5) such that $d(a_i^\phi, a_i^{'\phi}) \leq \mathsf{d}$, the number of different elements in the set S_i obtained from a_i^ϕ,$a_i^{'\phi}$ at Line 6 is at most $\mathsf{d} \cdot \hat{d}_i^\phi$. Therefore, the restricted sensitivity of the function f (denoted by $RS_f\left(\mathcal{H}_{\hat{d}_i^\phi}\right)$in Definition 5) is not larger than $\mathsf{d} \cdot \hat{d}_i^\phi \cdot \frac{e^{\varepsilon_1}+1}{e^{\varepsilon_1}-1} \cdot \frac{1}{\mathsf{d}} = \hat{d}_i^\phi \cdot \frac{e^{\varepsilon_1}+1}{e^{\varepsilon_1}-1}$. Hence, by Definition 5, the publication of $\tilde{t}_i$ at Line 8 is ε_2-edge local differentially private. ◀

We now discuss the accuracy of our estimation and its relation with the parameter ζ appearing at Line 4 the algorithm. We will see that ζ controls the trade off between the bias and the accuracy. The smaller ζ is, the smaller the average noise gets, but the larger the probability of bias and its expected magnitude is.

In the following lemma, we discuss that the projection $\mu_{\hat{d}_i}^\phi$ applied at Line 5 changes the adjacency vector a_i^ϕ only with small probability.

▶ **Lemma 21.** *For any $\zeta > 0$, with probability at least $1 - \zeta$, $|\tilde{d}_i - d_i| < (\ln \frac{n}{\zeta})/\varepsilon_0$ for all i.*

Proof. Using the cumulative distribution function of the Laplacian random variable, we have $\mathbb{P}\left[|\tilde{d}_i - d_i| \geq \varepsilon_0 \ln \frac{n}{\zeta}\right] \leq \frac{\zeta}{n}$. Thus, by taking this inequality for all $i \in [1, n]$, and using the union bound, we obtain $\mathbb{P}\left[\exists i \in [1, n], |\tilde{d}_i - d_i| \geq \varepsilon_0 \ln \frac{n}{\zeta}\right] \leq \zeta$. ◀

We show that our estimation has no bias with high probability in the subsequent theorem.

▶ **Theorem 22.** *With probability at least $1 - \zeta$, algorithm 2 provides an unbiased estimate of the number of triangles in the graph, i.e. $\mathbb{E}\left[\hat{f}_\triangle(G)\right] = \#C_3(G)$.*

Proof. As discussed in Definition 7, we have that $\mathbb{E}(\hat{a}_{j,k}^\phi) = a_{j,k}^\phi$. Using Lemma 21, with probability at least $1 - \zeta$, $\hat{d}_i^\phi$ is larger than d_i^ϕ for all $i \in [1, n]$, and the function $\mu_{\hat{d}_i^\phi}$ has no effect. Consequently, S_i precisely represents the set of forks centered on node ν_i, encompassing all possible triangles. Therefore, $\hat{t}_i$ is an unbiased estimate of the number of triangles (ν_i, ν_j, ν_k) such that $j < i < k$. Given that Laplace noise is centered and triangles can be decomposed accordingly, $\hat{f}_\triangle(G)$ is an unbiased estimation of $f_\triangle(G)$. ◀

Corollary 23 ensures that even in the unlikely event of some clipping occurring, the resulting bias would still represent only a small fraction of the actual count.

▶ **Corollary 23.** *The expected value of the bias of Algorithm 2 is bounded by $\mathcal{O}\left(\frac{\zeta}{\varepsilon_0 n}\#C_3\right)$.*

Proof. When the corrected estimated degree $\hat{d}_i^\phi$ is smaller than the actual degree d_i, $d_i - \hat{d}_i^\phi$ edges are excluded. This exclusion introduces a bias because the potential triangles involving these excluded edges are not counted. For each user i and their neighbor j, let $t_i^{(j)}$ denote the number of triangles counted by user i that involve the edge (ν_i, ν_j). We also define $t_i^{\max} = \max_j t_i^{(j)}$. Then, the maximum bias resulting from a single clipped edge can be bounded by $t_i^{\max}$.

The expected number of clipped edges for user i is determined by evaluating the following integral, where $\beta = \frac{\ln(n/\zeta)}{\varepsilon_0}$ serves as the correction term for the degree:

$$\int_{-\infty}^{-\beta} \frac{\varepsilon_0}{2} e^{-\varepsilon_0 |x|}(-x-\beta)\, dx = -\frac{\varepsilon_0}{2}\left[\frac{x-\beta}{\varepsilon_0} e^{-\varepsilon_0 x} + \frac{1}{\varepsilon_0^2} e^{-\varepsilon_0 x}\right]_{\beta}^{\infty} = \frac{1}{2\varepsilon_0} e^{-\varepsilon_0 \beta} = \frac{\zeta}{2\varepsilon_0 n}.$$

We obtain the final result by combining these elements and observing that $\sum_i t_i^{\max} \leq \sum_{i,j} t_i^{(j)} \leq 2 f_{\triangle}(G)$. ◀

The accuracy of our estimation is demonstrated in the subsequent theorem.

▶ **Theorem 24.** *When $\zeta \leq \varepsilon_0$, the squared expected ℓ_2-error of algorithm 2 is bounded by*

$$\mathcal{O}\left(\frac{\delta^3 n}{\varepsilon_1^2} + \frac{\delta d_{max} n}{\varepsilon_1^2 \varepsilon_2^2} + \frac{n \ln^2(n/\zeta)}{\varepsilon_0^2 \varepsilon_1^2 \varepsilon_2^2}\right).$$

Proof. The squared ℓ_2-error can be decomposed into the square of the bias plus the variance. We have established in Corollary 23 that the bias of the algorithm is bounded by $\mathcal{O}\left(\frac{\zeta}{\varepsilon_0 n} \#C_3\right) = \mathcal{O}\left(\delta^2\right)$. We will now focus on bounding the variance of the algorithm. This variance arises from two distinct sources: the randomized response query and the Laplacian query with restrictive sensitivity.

Regarding the noise introduced by the Laplacian query with restrictive sensitivity, its variance is simply the sum of the variances of each term, which is

$$9\left(\frac{e^{\varepsilon_1}+1}{e^{\varepsilon_1}-1}\right)^2 \sum_{\nu_i \in V} \frac{\hat{d}_i^2}{\varepsilon_2^2} = \mathcal{O}\left(\frac{\varepsilon_0^2 \delta d_{max} n + n \ln^2(n/\zeta)}{\varepsilon_2^2 \varepsilon_1^2 \varepsilon_0^2}\right).$$

Next, we consider the variance from the randomized response query. In the following equations, we use the notation $\mathcal{N}_{j,k}^*$ to denote the set of neighbors ν_i of both ν_j and ν_k such that $j < i < k$. Note that by including one node from $\mathcal{N}_{j,k}^*$ along with ν_j and ν_k, a triple in S_2^* is formed. Similarly, including two nodes from $\mathcal{N}_{j,k}^*$ along with ν_j and ν_k results in a quadruplet in $\#C_4^*$. We also notice from Definition 7 that, for $(j,k) \neq (j',k')$, $\hat{a}_{j,k}^{\phi}$ is independent to $\hat{a}_{j',k'}^{\phi}$ and $\mathrm{Cov}\left(\hat{a}_{j,k}^{\phi}, \hat{a}_{j',k'}^{\phi}\right) = 0$. Hence,

$$\begin{aligned}\mathrm{Var}\left(\sum_{\nu_i \in V^\phi} \sum_{(j,k)\in S_i} \hat{a}_{j,k}^{\phi}\right) &= \sum_{(\nu_j,\nu_k)\in (V^\phi)^2}\left[\sum_{\nu_i \in \mathcal{N}_{j,k}^*} \mathrm{Var}\left(\hat{a}_{j,k}^{\phi}\right) + \sum_{\nu_i,\nu_{i'} \in \mathcal{N}_{j,k}^*} \mathrm{Cov}\left(\hat{a}_{j,k}^{\phi}, \hat{a}_{j,k}^{\phi}\right)\right] \\ &= \mathcal{O}\left((S_2^* + \#C_4^*)/\varepsilon_1^2\right)\end{aligned}$$

By Theorem 13 and Collorary 18, $\mathrm{Var}\left(\hat{f}(G)\right) = \mathcal{O}\left(\frac{\delta^3 n}{\varepsilon_1^2} + \frac{\delta d_{max} n}{\varepsilon_1^2 \varepsilon_2^2} + \frac{n \ln^2(n/\zeta)}{\varepsilon_0^2 \varepsilon_1^2 \varepsilon_2^2}\right)$. ◀

In the previous work [16], the number of terms in the variance calculation is bounded by the number of cycles of length four, which is $\mathcal{O}\left(d_{\max}^3 n\right)$. We reduce that number to $\#C_4^* = \mathcal{O}\left(\delta^3 n\right)$ using the `GetOrdering` function in Line 2 and by including only pairs (j,k) such that $j < i < k$. It is known that $\delta \leq d_{\max}$ and, in many practical graphs, the degeneracy is much smaller than the maximum degree.

5 Odd Length Cycle Counting

In this section, we will describe how to utilize low-degree ordering to accurately count odd-length cycles in graphs with bounded degeneracy. Some concepts are extended from the previous section. As shown in Algorithm 3, the algorithm for estimating the number of odd-length cycles is similar to Algorithm 2, except that the restricted sensitivity at Line 9 is larger, and at Line 8, we replace $\hat{a}^{\phi}_{i,j}$ with an estimate for the number of paths under specific constraints. We discuss the privacy of the algorithm in the subsequent theorem. The main challenge of the proof is to demonstrate that the Laplacian query under restricted sensitivity at Lines 5-9 is ε_2-differentially private.

Algorithm 3 Our algorithm for estimating the number of odd-length cycles in degeneracy-bounded graphs.

```
1  Function OddCycleCounting
      Input: Graph $G = (V, E)$, privacy budget $\varepsilon = \varepsilon_0 + \varepsilon_1 + \varepsilon_2$, $k$ an odd number not
             smaller than 5, parameter $\zeta$
      Output: Estimation of the number of $k$-cycles in $G$
2     [All Users and Server] $\phi \leftarrow$ GetOrdering$(G, \varepsilon_0)$ (Algorithm 1);
3     [All Users and Server] Inquire the unbiased randomized response query with privacy
        budget $\varepsilon_1$ to all users.
4     [Server] Let $(\hat{a}^{\phi}_{i,j})$ represent the results collected from this query. The server then
        distributes $(\hat{a}^{\phi}_{i,j})$ to all users.
5     [Server] Calculate
           $$\#\hat{P}_{k-4} := \sum_{(l_1,\ldots,l_{k-3}) \in V^{k-3}} \prod_{q \in [1,k-4]} \hat{a}^{\phi}_{l_q,l_{q+1}},$$
        then send this information to all users;
6     [User $i$] $\hat{d}^{\phi}_i \leftarrow \tilde{d}^{\phi}_i + \frac{1}{\varepsilon_0} \ln(n/\zeta)$;
7     [User $i$] $a^{\phi}_i \leftarrow \mu_{\hat{d}^{\phi}_i}(a^{\phi}_i)$;
8     [User $i$] $S_i \leftarrow \{(j, \kappa) \mid a^{\phi}_{i,j} = a^{\phi}_{i,\kappa} = 1, j < i < \kappa\}$;
9     [User $i$] $\hat{c}_i \leftarrow \sum_{(j,\kappa) \in S_i} \#\hat{P}^{(i)}_{k-2}(j, \kappa)$ when
           $$\#\hat{P}^{(i)}_{k-2}(j, \kappa) = \sum_{(l_1,\ldots,l_{k-1}) \in X^{(i)}_{k-2}(j,\kappa)} \prod_{q \in [1,k-2]} \hat{a}^{\phi}_{l_q,l_{q+1}}$$
        and $X^{(i)}_{k-2}(j, \kappa)$ is a set of non-repeating combination of $k - 1$ vertices in $G^{\phi}$ with
        endpoints $\nu_j$ and $\nu_\kappa$, such that, for any three consecutive nodes $(\nu_q, \nu_r, \nu_s)$ in the path
        with monotonic ranks, the node $\nu_i$ has a lower rank than $\nu_r$;
10    [User $i$] $\tilde{c}_i \leftarrow \hat{c}_i + \mathtt{Lap}\left(3 \cdot \left(\frac{e^{\varepsilon_1}+1}{e^{\varepsilon_1}-1}\right)^2 \cdot \hat{d}^{\phi}_i \cdot \#\hat{P}_{k-4}/\varepsilon_2\right)$;
11    [User $i$] Upload $\tilde{c}_i$ to the central server;
12    [Server] $\hat{f}_k(G) \leftarrow \sum_{\nu_i \in V} \tilde{c}_i$;
13    return $\hat{f}_k(G)$;
```

▶ **Theorem 25.** *Algorithm 3 provides $(\varepsilon_0 + \varepsilon_1 + \varepsilon_2)$-edge local differential privacy.*

Proof. We need to demonstrate that Lines 5-9 of the algorithm, involving the Laplacian query with restricted sensitivity on $\mathcal{H}_{\hat{d}^{\phi}_i}$, ensure ε_2-differential privacy. Following the arguments of Theorem 20, we assert that altering d entries of $a^{\phi}_{i,j}$ changes the set S_i by at most $\mathsf{d} \cdot \hat{d}^{\phi}_i$ elements. A single element change in S_i can alter the value of $\hat{c}_i$ by

$$\#\hat{P}^{(i)}_{k-2}(j,\kappa) = \sum_{(l_1,\ldots,l_{k-1})\in X^{(i)}_{k-2}(j,\kappa)} \prod_{q\in[1,k-2]} \hat{a}^{\phi}_{l_q,l_{q+1}} \leq \left(\frac{e^{\varepsilon}+1}{e^{\varepsilon}-1}\right)^2 \#\hat{P}_{k-4}.$$

Therefore, the restricted sensitivity of $\tilde{c}_i$ is $\mathsf{d}\cdot\hat{d}^{\phi}_i\left(\frac{e^{\varepsilon}+1}{e^{\varepsilon}-1}\right)^2\#\hat{P}_{k-4}/\mathsf{d} = \hat{d}^{\phi}_i\left(\frac{e^{\varepsilon}+1}{e^{\varepsilon}-1}\right)^2\#\hat{P}_{k-4}$. Consequently, the publication of $\tilde{c}_i$ at Line 9 is ε_2-differentially private. ◀

The bias of the algorithm is given in the following theorem.

▶ **Theorem 26.** *With a probability of at least $1-\zeta$, Algorithm 3 provides an unbiased estimate of the number of k-cycles in any graph G for any odd integer k.*

Proof. Since we publish $\hat{a}^{\phi}_{l_q,l_{q+1}}$ using the unbiased randomized response query, the publication is an unbiased estimation of $a^{\phi}_{l_q,l_{q+1}}$. Furthermore, as those estimators are independent from one another, for each $(j,\kappa)\in S_i$ and $\{l_1,\ldots,l_{k-1}\}\in X^{(i)}_{k-2}(j,\kappa)$, $\prod_{q\in[1,k-2]}\hat{a}^{\phi}_{l_q,l_{q+1}}$ is an unbiased estimate of $\prod_{q\in[1,k-2]}a^{\phi}_{l_q,l_{q+1}}$. It results from this that $\#\hat{P}^{(i)}_{k-2}(j,\kappa)$ is an unbiased estimator of the number of paths between j and κ with length $k-2$ such that, for any three consecutive nodes (ν_q,ν_r,ν_s) with monotonic ranks, the node ν_i has a lower rank that ν_r. We denote the number of such paths as $\#P^{(i)}_{k-2}(j,\kappa)$.

Let us introduce $C^{(i)}_k = \sum_{(j,k)\in S_i}\#P^{(i)}_{k-2}(j,\kappa)$. Assuming no clipping occurs, which happens with a probability of at least $1-\zeta$, we have by linearity of expectation that both $\hat{c}_i$ and $\tilde{c}_i$ are unbiased estimators of $C^{(i)}_k$. Therefore, all that remains to be proven is that the number of k-cycles in G is equal to $\sum_{\nu_i\in V^{\phi}}C^{(i)}_k$. It is evident that for each element counted in $\sum_{\nu_i\in V^{\phi}}C^{(i)}_k$, there is a corresponding cycle $(\nu_i,l_1,\ldots,l_{k-1})$ in G^{ϕ} and, also, in G.

Conversely, consider a cycle of length k in G. Since it is also a cycle in G^{ϕ}, we can represent it in G^{ϕ} as $(\nu_1,\ldots,\nu_k)$. Because the cycle is of odd length, there exist three consecutive nodes with a monotonic rank. Among all possible triplets, consider the one where the central node has the smallest rank, denoted as (ν_j,ν_i,ν_κ) with $j<i<\kappa$. Furthermore, let $j=l_1$ and $\kappa=l_{k-1}$, and assign the indices of the other nodes in the cycle to l_2 through l_{k-2} in the order they appear in the cycle. Thus, the cycle is counted in $\sum_{\nu_i\in V^{\phi}}C^{(i)}_k$. Furthermore, if any other node in the cycle were chosen as ν_i, the remaining path would not be part of $X^{(i)}_{k-2}(j,\kappa)$. This ensures that each cycle is counted exactly once in $\sum_{\nu_i\in V^{\phi}}C^{(i)}_k$. ◀

Finally, the ℓ_2-error of Algorithm 3 is proven in the next theorem. The most challenging aspect of this theorem is to bound the covariance in the summation at Lines 8 and 10. We assert that any two dependent elements of $X^{(i)}_{k-2}(j,\kappa)$ can be considered as a set containing an even number of edges which forms multiple disjoint cycles with specific properties. Consequently, we can utilize our results from Corollary 19 to bound the number of such pairs. The proof of the theorem is given in the appendix of this paper.

▶ **Theorem 27.** *When $\zeta\leq\varepsilon_0$, the expected squared ℓ_2-error of algorithm 3 is bounded by*

$$\mathcal{O}\left(\frac{\delta^3}{\varepsilon_1^2}\left(\frac{1}{\varepsilon_1^2}+\delta\right)^{k-3}n^{k-2}+\frac{\delta^{k-2}d_{max}n^{k-2}}{\varepsilon_2^2\varepsilon_1^4}+\frac{\delta^{k-3}n^{k-2}\ln^2(n/\zeta)}{\varepsilon_2^2\varepsilon_1^4\varepsilon_0^2}\right).$$

Before proving Theorem 27, we demonstrate the following lemma.

▶ **Lemma 28.** *The expected value of the bias of Algorithm 3 is bounded by $\mathcal{O}\left(\frac{\zeta}{\varepsilon_0 n}\#C_k\right)$.*

Proof. We have already seen the proof of Corollary 23 that the expected value of the number of clipped edges for user i was bounded by $\frac{\zeta}{2\varepsilon_0 n}$. We now have to bound the bias created by one edge removal, i.e. the maximal number of cycles one edge can part of.

With $c_i^{(j)}$ the number of cycles counted by i that involve edge (i, j), the maximal bias for user i is bounded by $\sum_j c_i^{(j)}$, and the bias of the algorithm by $\frac{\zeta}{2\varepsilon_0 n} \sum_{i,j} c_i^{(j)} \leq \mathcal{O}\left(\frac{\zeta}{\varepsilon_0 n} \# C_k\right)$. ◀

Now, we are ready to prove Theorem 27.

Proof of Theorem 27. The squared ℓ_2-error can be decomposed into the square of the bias plus the variance. In Lemma 28, we established that the bias of the algorithm is bounded by $\mathcal{O}\left(\frac{\zeta}{\varepsilon_0 n} \# C_k\right) = \mathcal{O}\left(\delta^{\frac{k+1}{2}} n^{\frac{k-3}{2}}\right)$. We will now focus on bounding the variance of the algorithm.

Let the indicator variable $\mathbb{1}_{(l_1,\ldots,l_p)}$ be 1 if the path $(\nu_{l_1}, \ldots, \nu_{l_p})$ exists in G^ϕ, and 0 otherwise. We also denote the random variable $\prod_{q\in[1,p]} \hat{a}^{\phi}_{l_q,l_{q+1}}$ by $Z_{(l_1,\ldots,l_{p+1})}$. Finally, we define $U_{(l_1,\ldots,l_{p+1})} = Z_{(l_1,\ldots,l_{p+1})} - \mathbb{1}_{(l_1,\ldots,l_{p+1})}$. This random variable $U_{(l_1,\ldots,l_{p+1})}$ has the properties that $\mathbb{E}\left[U_{(l_1,\ldots,l_{p+1})}\right] = 0$ and $\mathrm{Var}\left(U_{(l_1,\ldots,l_{p+1})}\right) = \mathrm{Var}\left(Z_{(l_1,\ldots,l_{p+1})}\right)$.

Similar to the case with triangles, the variance of Algorithm 3 arises from both the unbiased randomized response query and the Laplacian query with restricted sensitivity.

Concerning the variance term coming from the randomized response, we have to compute the variance of

$$\hat{C} = \sum_{\nu_i \in V} (\hat{c}_i - c_i) = \sum_{\nu_i \in V} \sum_{(j,\kappa)\in S_i} \sum_{\{l_1,\ldots,l_{k-1}\}\in X^{(i)}_{k-2}(j,\kappa)} U_{(l_1,\ldots,l_{k-1})}.$$

We have to take into account the term that comes from the sum of the variances of the U as well as the one coming from the covariances between them.

To compute the sum of variances, we start with:

$$\mathrm{Var}\left(U_{(l_1,\ldots,l_{k-1})}\right) = \prod_{q\in[1,k-2]} \mathrm{Var}\left(\hat{a}^{\phi}_{l_q,l_{q+1}}\right) = \mathcal{O}\left(\frac{1}{\varepsilon_1^{2k-4}}\right).$$

Additionally, for each i and $(j,\kappa) \in S_i$, the cardinality of $X^{(i)}_{k-2}(j,\kappa)$ is bounded by n^{k-3}, and the number of ways to choose (i, j, κ) is bounded by S_2^*, which is $\mathcal{O}\left(\delta^2 n\right)$ by Theorem 13. This contributes a term in the variance from the sum of variances bounded by $\mathcal{O}\left(\delta^2 n^{k-2}/\varepsilon_1^{2k-4}\right)$.

To analyze the term arising from the covariances, we first examine the covariance between $U_{(l_1,\ldots,l_{k-1})}$ and $U_{(l'_1,\ldots,l'_{k-1})}$. In the following equations, let A be the set of edges that appear only in $(l_1, \ldots, l_{k-1})$ or $(l'_1, \ldots, l'_{k-1})$, and let B be the set of edges that appear in both. Recall that, for any (i, j) $\mathbb{E}\left[\hat{a}^{\phi}_{i,j}\right] = 0$ and $\mathbb{E}\left[\hat{a}^{\phi}_{i,j}\right] = \mathrm{Var}\left(\hat{a}^{\phi}_{i,j}\right)$.

$$\begin{aligned}
&\mathrm{Cov}\left(U_{(l_1,\ldots,l_{k-1})}, U_{(l'_1,\ldots,l'_{k-1})}\right) \\
&= \mathbb{E}\left[\prod_{q\in[1,k-2]} \hat{a}^{\phi}_{l_q,l_{q+1}} \prod_{q\in[1,k-2]} \hat{a}^{\phi}_{l'_q,l'_{q+1}}\right] - \mathbb{1}_{(l_1,\ldots,l_{k-1})}\mathbb{1}_{(l'_1,\ldots,l'_{k-1})} \\
&= \prod_{(i,j)\in A} \mathbb{1}_{(i,j)} \prod_{(i,j)\in B} \mathrm{Var}\left(a^{\phi}_{i,j}\right) - \prod_{q\in[1,k-2]} \mathbb{1}_{(l_q,l_{q+1})}\mathbb{1}_{(l'_q,l'_{q+1})}.
\end{aligned}$$

We observe that the covariance between $U_{(l_1,\ldots,l_{k-1})}$ and $U_{(l'_1,\ldots,l'_{k-1})}$ is zero if the paths $(\nu_{l_1},\ldots,\nu_{l_{k-1}})$ and $(\nu_{l'_1},\ldots,\nu_{l'_{k-1}})$ do not share at least one common edge or if the edges present in only one of the paths are not present in the original graph. Now consider the situation where the covariance is non-zero. We have that $|B| > 0$. Additionally, we will denote ν_i the node responsible for counting this instance of $U_{(l_1,\ldots,l_{k-1})}$ and $\nu_{i'}$ the one responsible for $U_{(l'_1,\ldots,l'_{k-1})}$.

Let $\mathsf{V} := \{\nu_i, \nu_{l_1}, \ldots, \nu_{l_{k-1}}, \nu_{i'}, \nu_{l'_1}, \ldots, \nu_{l'_{k-1}}\}$, and let

$$\mathsf{E} := \bigcup_{1\le q\le k-2} \{(\nu_{l_q}, \nu_{l_{q+1}}), (\nu_{l'_q}, \nu_{l'_{q+1}})\} \cup \{(\nu_{l_{k-1}}, \nu_i), (\nu_i, \nu_{l_1}), (\nu_{l'_{k-1}}, \nu_{i'}), (\nu_i, \nu_{l'_1})\}.$$

In other words, the set E consists of the edges in the paths $\{l_1, \ldots, l_{k-1}\}$ and $\{l'_1, \ldots, l'_{k-1}\}$, along with the additional edges $(\nu_{l_{k-1}}, \nu_i)$, (ν_i, ν_{l_1}), $(\nu_{l'_{k-1}}, \nu_{i'})$, and $(\nu_i, \nu_{l'_1})$. Additionally, let

$$\begin{aligned} A' \quad &:= A \cup \{(\nu_{l_{k-1}}, \nu_i), (\nu_{l'_{k-1}}, \nu_{i'}) \mid (\nu_{l_{k-1}}, \nu_i) \neq (\nu_{l'_{k-1}}, \nu_{i'})\} \\ &\cup \{(\nu_i, \nu_{l_1}), (\nu_{i'}, \nu_{l'_1}) \mid (\nu_i, \nu_{l_1}) \neq (\nu_{i'}, \nu_{l'_1})\}. \end{aligned}$$

Similarly, let

$$B' \quad := B \cup \{(\nu_{l_{k-1}}, \nu_i) \mid (\nu_{l_{k-1}}, \nu_i) = (\nu_{l'_{k-1}}, \nu_{i'})\} \cup \{(\nu_i, \nu_{l_1}) \mid (\nu_i, \nu_{l_1}) = (\nu_{i'}, \nu_{l'_1})\}.$$

In other words, the sets A' and B' are the sets A and B extended to include the additional edges (ν_i, ν_{l_1}), $(\nu_{l_{k-1}}, \nu_i)$, $(\nu_{i'}, \nu_{l'_1})$, and $(\nu_{l'_{k-1}}, \nu_{i'})$.

We introduce $\mathbf{d}$ the difference between the cardinal of B' and B, $\mathbf{d} := |B'| - |B|$. Let $q \in [1, k-2]$ be the cardinality of B. In this case, the covariance is $\mathcal{O}\left(1/\varepsilon_1^{2q}\right)$. We have that $|A'| + 2|B'| = 2k$, which gives $|A'| = 2k - 2q - 2\mathbf{d}$.

In the next step, we will calculate the number of the pairs of paths with $|A'| = 2(k-q-\mathbf{d})$. Let us consider the degree of each node in (V, E). It is clear that the degrees are neither greater than four nor less than two. A node has a degree of three only if one of the three edges incident to it belongs to B' and the other two to A'. A node has a degree of four if all four edges incident to it are in A', and it has a degree of two if both edges incident to it are either in A' or in B'. Hence, if we consider the graph (V, A'), we have a graph of degree two or four, which is a union of multiple disjoint cycles.

Let the number of those disjoint cycles be c and the size of those cycles be $r_1, \ldots r_c$. We have that $\sum_{t=1}^{c} r_t = 2k - 2q - 2\mathbf{d}$, i.e. $(r_1, \ldots, r_c)$ is a partition of $2k - 2q - 2\mathbf{d}$. We know that the number of such partitions is bounded by a function of k. Let suppose that the bound is $f(k)$.

Let us give the number of A' with cycle size $(r_1, \ldots, r_c)$. We can use Corollary 16 to show that the number of such sets A' is $\mathcal{O}\left(\prod_{t=1}^{c} \delta^{r_t/2} n^{r_t/2}\right) = \mathcal{O}\left(\delta^{k-q-\mathbf{d}} n^{k-q-\mathbf{d}}\right)$. When $\mathbf{d} = 0$, we know that $\{\nu_i, \nu_j\}$ and $\{\nu_i, \nu_k\}$ are in A'. There are three consecutive nodes with monotonic ranks in the union of disjoint cycles (V, A'). Hence, we can use Corollary 19 to show that the number of such sets A' is bounded by $\mathcal{O}\left(\delta^{k-q+1} n^{k-q-1}\right)$. By combining the two cases, we can conclude that the number of possible sets A' with cycle size $(r_1, \ldots, r_c)$ is at most $\mathcal{O}\left(\delta^{k-q+1-\mathbf{d}} n^{k-q-1}\right)$. The number of possible A' is then $f(k) \cdot \mathcal{O}\left(\delta^{k-q+1-\mathbf{d}} n^{k-q-1}\right)$. As k is a constant, the number is $\mathcal{O}\left(\delta^{k-q+1-\mathbf{d}} n^{k-q-1}\right)$.

We then consider the number of configurations for B', which consists of a union of disjoint paths. Let the number of paths be c and their lengths be $r_1, \ldots, r_c$. We have that $|r_1| + \cdots + |r_c| = q$, and $(r_1, \ldots, r_c)$ forms a partition of q. The number of possible partitions is bounded by a function of k, denoted as $f(k)$. Each part must begin and end in the node set A', where $|A'| \le 2k$. Therefore, the number of possible paths r_t is at most $4k^2 n^{r_t - 1}$, and the number of possible sets B' with the partition $(r_1, \ldots, r_c)$ is at most $\prod_{t=1}^{c} 4k^2 n^{r_t-1} = \mathcal{O}\left(n^{q-1}\right)$. Hence, the total number of possible sets B' is $f(k) \cdot \mathcal{O}\left(n^{q-1}\right) = \mathcal{O}\left(n^{q-1}\right)$.

Consequently, for each set A', the number of possible configurations for B' is at most $\mathcal{O}\left(n^{q-1}\right)$. The number of pairs of paths $\{l_1, \dots, l_{k-1}\}$ and $\{l'_1, \dots, l'_{k-1}\}$ with $|A'| = 2(k - q - \mathbf{d})$ is then at most $\mathcal{O}\left(\delta^{k-q+1-\mathbf{d}} n^{k-q-1} \cdot n^{q-1}\right) = \mathcal{O}\left(\delta^{k-q+1} n^{k-2}\right)$. Each of these pairs contributes $\mathrm{Var}\left(\hat{a}^{\phi}_{j,k}\right)^{2q} = \mathcal{O}\left(1/\varepsilon_1^{2q}\right)$ to the covariance sum.

The covariance of $\hat{C}$ can then be calculated as follows:

$$\mathcal{O}\left(\sum_{q=1}^{k-2} \delta^{k-q+1} n^{k-2} \frac{1}{\varepsilon_1^{2q}}\right) = \mathcal{O}\left(\frac{n^{k-2}\delta^3}{\varepsilon_1^2}\left(\delta + \frac{1}{\varepsilon_1^2}\right)^{k-3}\right).$$

Since this bound is larger than the one for the sum of variances, we can disregard the latter.

To compute the variance resulting from the Laplacian query with restricted sensitivity, we sum the variance of the Laplacian distribution for all nodes:

$$9\left(\frac{e^{\varepsilon}+1}{e^{\varepsilon}-1}\right)^4 \mathbb{E}\left[\#\hat{P}^2_{k-4}\right] \sum_{\nu_i \in V} \frac{\hat{d}_i^2}{\varepsilon_2^2} = \mathcal{O}\left(\frac{\delta d_{max} n}{\varepsilon_2^2 \varepsilon_1^4} + \frac{n \ln^2(n/\zeta)}{\varepsilon_2^2 \varepsilon_1^4 \varepsilon_0^2}\right) \mathbb{E}\left[\#\hat{P}^2_{k-4}\right]. \tag{1}$$

Let us now consider the expected value

$$\mathbb{E}\left[\#\hat{P}^2_{k-4}\right] = \mathbb{E}\left[\#\hat{P}_{k-4}\right]^2 + \mathrm{Var}\left(\#\hat{P}_{k-4}\right) = \#P^2_{k-4} + \mathrm{Var}\left(\#\hat{P}_{k-4}\right). \tag{2}$$

By Lemma 14 and the fact that $k-4$ is an odd number, we have $\#P^2_{k-4} = \mathcal{O}\left(\delta^{k-3} n^{k-3}\right)$. The variance can be decomposed into the sum of the variances of each path, which is bounded by $\mathcal{O}\left(n^{k-3}/\varepsilon_1^{2k-8}\right)$, and the sum of covariances.

The covariance is non-zero only if at least two edges are shared between the two paths and all edges that appear only once exist in the original graph. As previously discussed, this forms a cycle structure, except for the path extremities that do not need to be connected. Recall the definitions of the sets A and B from the previous paragraph.

The set A consists of two paths at the extremities and multiple disjoint cycles. Suppose the number of edges in A is $2p$, the number of edges in the two paths are q_1 and q_2, and the number of disjoint cycles is c, with the number of edges in these cycles being $r_1, \dots, r_c$. This gives us $2p = q_1 + q_2 + \sum_{i=1}^{c} r_i$. In other words, $(q_1, q_2, r_1, \dots, r_c)$ forms a partition of $2p \leq 2k$. The number of such partitions is bounded by a function of k. Let the bound be $f(k)$.

We now discuss the number of possible configurations of A for the partition $(q_1, q_2, r_1, \dots, r_c)$. From Lemma 14 and Corollary 16, the number of cycles of length q is bounded by $\mathcal{O}\left(\delta^{q/2} n^{q/2}\right)$, and the number of paths of length q is bounded by $\mathcal{O}\left(\delta^{q/2} n^{q/2+1}\right)$. Thus, the number of configurations for the partition $(q_1, q_2, r_1, \dots, r_c)$ is:

$$\mathcal{O}\left(\delta^{q_1/2} n^{q_1/2+1} \cdot \delta^{q_2/2} n^{q_2/2+1} \cdot \prod_{t=1}^{c} \delta^{r_t/2} n^{r_t/2}\right) = \mathcal{O}\left(\delta^p n^{p+2}\right).$$

Hence, the number of possible configurations for A with $2p$ edges is no more than $f(k) \cdot \mathcal{O}\left(\delta^p n^{p+2}\right) = \mathcal{O}\left(\delta^p n^{p+2}\right)$.

The number of edges in B is $(2k - 8 - 2p)/2 = k - p - 4$. Using the previous argument when calculating the number of possible set B', we obtain that the number of configurations for B is $\mathcal{O}\left(n^{k-p-5}\right)$. The number of configurations with $|A| = 2p$ is then $\mathcal{O}\left(\delta^p n^{p+2} \cdot n^{k-p-5}\right) = \mathcal{O}\left(\delta^p n^{k-3}\right)$. Hence, the overall number of combinations is $\sum_{p=1}^{k-5} \mathcal{O}\left(\delta^p n^{k-3}\right) = \mathcal{O}\left(\delta^{k-5} n^{k-3}\right)$.

From the previous paragraph, we observe that the covariance term outweighs the sum of the variances, leading to $\mathrm{Var}\left(\#\hat{P}_{k-4}\right) = \mathcal{O}\left(\delta^{k-5}n^{k-3}\right)$. Additionally, when calculating $\mathbb{E}\left[\#\hat{P}^2_{k-4}\right]$ in (2), it is evident that $\#P^2_{k-4}$ dominates $\mathrm{Var}\left(\#\hat{P}_{k-4}\right)$, resulting in $\mathbb{E}\left[\#\hat{P}^2_{k-4}\right] = \mathcal{O}\left(\delta^{k-3}n^{k-3}\right)$. Substituting $\mathbb{E}\left[\#\hat{P}^2_{k-4}\right]$ with $\mathcal{O}\left(\delta^{k-3}n^{k-3}\right)$ in (1), we find that the variance from the Laplacian mechanism is bounded by

$$\mathcal{O}\left(\frac{\delta^{k-2}d_{max}n^{k-2}}{\varepsilon_2^2\varepsilon_1^4} + \frac{\delta^{k-3}n^{k-2}\ln^2(n/\zeta)}{\varepsilon_2^2\varepsilon_1^4\varepsilon_0^2}\right).$$

We obtain the theorem result by summing the variance from the unbiased randomized response query and the variance from the Laplacian query with restricted sensitivity. ◀

6 Conclusion

In this work, we introduced a private vertex ordering algorithm. The transformation on the graph induced by this ordering reduces the count of specific order-sensitive motifs while preserving the overall graph structure. Due to its reliance on the Laplacian mechanism, the algorithm performs well even in high-privacy settings, making it an excellent preprocessing step for subgraph counting queries.

Within this framework, we first propose a new triangle counting algorithm whose accuracy depends on the count of specific ordered subgraphs. By combining this algorithm with the ordering preprocessing step, we achieve an expected error of $\mathcal{O}(n)$ for graphs with bounded degeneracy, compared to the $\mathcal{O}\left(n^2\right)$ error seen in the current state of the art.

Subsequently, we extended the algorithm to address the more general case of odd-length cycle counting. We propose the first purely local differentially private counting algorithm specifically designed for cycles longer than triangles. Under the assumption of bounded degeneracy, the algorithm achieves an error of $\mathcal{O}\left(n^{(k-1)/2}\right)$ for cycles of length k.

Due to the constraints of local differential privacy, it might be assumed that the range of tasks we can perform on graphs under this privacy notion is limited. However, in this work, we demonstrate that more precise information can be published under local differential privacy by restricting our inputs to certain types of graphs. We believe that parameterized algorithms under local differential privacy represent an intriguing research area that can contribute significantly to both algorithm design and information privacy.

One limitation of this method is that the relative error can become significantly large when the number of cycles is small (or even zero), even in cases where the graph's degeneracy – and consequently the ℓ_2-error of our algorithm – is high. Identifying a class of graphs for which an algorithm with bounded relative error can be designed would be a direction for future research. Another question for future investigation is determining lower bounds for degeneracy-bounded graphs under the local differential privacy.

References

1 Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

2 Suman K. Bera, Lior Gishboliner, Yevgeny Levanzov, C. Seshadhri, and Asaf Shapira. Counting subgraphs in degenerate graphs. *ACM Journal of the ACM (JACM)*, 69(3):23:1–23:21, 2022. doi:10.1145/3520240.

3 Louis Betzer, Vorapong Suppakitpaisarn, and Quentin Hillebrand. Publishing number of walks and katz centrality under local differential privacy. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024. URL: https://openreview.net/forum?id=76UkTmdmkB.

4 Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. Differentially private data analysis of social networks via restricted sensitivity. In *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 87–96. ACM, 2013. doi:10.1145/2422436.2422449.

5 Marco Bressan. Faster algorithms for counting subgraphs in sparse graphs. *Algorithmica*, 83(8):2578–2605, 2021. doi:10.1007/s00453-021-00811-0.

6 Norishige Chiba and Takao Nishizeki. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14(1):210–223, 1985. doi:10.1137/0214017.

7 Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 1655–1658. ACM, 2018. doi:10.1145/3183713.3197390.

8 Damien Desfontaines and Balázs Pejó. Sok: Differential privacies. *Proc. Priv. Enhancing Technol.*, 2020(2):288–313, 2020. doi:10.2478/popets-2020-0028.

9 Laxman Dhulipala, George Z. Li, and Quanquan C. Liu. Near-optimal differentially private k-core decomposition. *CoRR*, abs/2312.07706, 2023. doi:10.48550/arXiv.2312.07706.

10 Laxman Dhulipala, Quanquan C. Liu, Sofya Raskhodnikova, Jessica Shi, Julian Shun, and Shangdi Yu. Differential privacy from locally adjustable graph algorithms: k-core decomposition, low out-degree ordering, and densest subgraphs. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 754–765. IEEE, 2022. doi:10.1109/FOCS54457.2022.00077.

11 Michael Dinitz, Satyen Kale, Silvio Lattanzi, and Sergei Vassilvitskii. Improved differentially private densest subgraph: Local and purely additive. *CoRR*, abs/2308.10316, 2023. doi:10.48550/arXiv.2308.10316.

12 Pål Grønås Drange, Patrick Greaves, Irene Muzi, and Felix Reidl. Computing complexity measures of degenerate graphs. In *18th International Symposium on Parameterized and Exact Computation, IPEC 2023, September 6-8, 2023, Amsterdam, The Netherlands*, volume 285 of *LIPIcs*, pages 14:1–14:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.IPEC.2023.14.

13 Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006. doi:10.1007/11787006_1.

14 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006. doi:10.1007/11681878_14.

15 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014. doi:10.1561/0400000042.

16 Talya Eden, Quanquan C. Liu, Sofya Raskhodnikova, and Adam D. Smith. Triangle counting with local edge differential privacy. In *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 52:1–52:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.52.

17 Alexandre V. Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, pages 211–222. ACM, 2003. doi:10.1145/773153.773174.

18 Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1106–1125. SIAM, 2010. doi:10.1137/1.9781611973075.90.

19 Michael Hay, Chao Li, Gerome Miklau, and David D. Jensen. Accurate estimation of the degree distribution of private networks. In *ICDM 2009, The Ninth IEEE International Conference on Data Mining, Miami, Florida, USA, 6-9 December 2009*, pages 169–178. IEEE Computer Society, 2009. doi:10.1109/ICDM.2009.11.

20 Quentin Hillebrand, Vorapong Suppakitpaisarn, and Tetsuo Shibuya. Communication cost reduction for subgraph counting under local differential privacy via hash functions. *CoRR*, abs/2312.07055, 2023. doi:10.48550/arXiv.2312.07055.

21 Quentin Hillebrand, Vorapong Suppakitpaisarn, and Tetsuo Shibuya. Unbiased locally private estimator for polynomials of laplacian variables. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 741–751. ACM, 2023. doi:10.1145/3580305.3599537.

22 Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. Locally differentially private analysis of graph statistics. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 983–1000. USENIX Association, 2021. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/imola.

23 Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. Communication-efficient triangle counting under local differential privacy. In *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 537–554. USENIX Association, 2022. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/imola.

24 Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. Differentially private triangle and 4-cycle counting in the shuffle model. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 1505–1519. ACM, 2022. doi:10.1145/3548606.3560659.

25 George Manoussakis. Listing all fixed-length simple cycles in sparse graphs in optimal time. In *Fundamentals of Computation Theory - 21st International Symposium, FCT 2017, Bordeaux, France, September 11-13, 2017, Proceedings*, volume 10472 of *Lecture Notes in Computer Science*, pages 355–366. Springer, Springer, 2017. doi:10.1007/978-3-662-55751-8_28.

26 Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 94–103. IEEE Computer Society, 2007. doi:10.1109/FOCS.2007.41.

27 Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.

28 Iyiola E. Olatunji, Thorben Funke, and Megha Khosla. Releasing graph neural networks with differential privacy guarantees. *Trans. Mach. Learn. Res.*, 2023, 2023. URL: https://openreview.net/forum?id=wk8oXR0kFA.

29 Zhan Qin, Ting Yu, Yin Yang, Issa Khalil, Xiaokui Xiao, and Kui Ren. Generating synthetic decentralized social graphs with local differential privacy. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 425–438. ACM, 2017. doi:10.1145/3133956.3134086.

30 Sofya Raskhodnikova and Adam D. Smith. Differentially private analysis of graphs. In *Encyclopedia of Algorithms*, pages 543–547. Springer, 2016. doi:10.1007/978-1-4939-2864-4_549.

31 Sina Sajadmanesh and Daniel Gatica-Perez. Locally private graph neural networks. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, pages 2130–2145. ACM, 2021. doi:10.1145/3460120.3484565.

32 Yue Wang, Xintao Wu, and Donghui Hu. Using randomized response for differential privacy preserving data collection. In *Proceedings of the Workshops of the EDBT/ICDT 2016 Joint Conference, EDBT/ICDT Workshops 2016, Bordeaux, France, March 15, 2016*, volume 1558 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016. URL: https://ceur-ws.org/Vol-1558/paper35.pdf.

33 Stanley L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

34 Qingqing Ye, Haibo Hu, Man Ho Au, Xiaofeng Meng, and Xiaokui Xiao. Towards locally differentially private generic graph metric estimation. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*, pages 1922–1925. IEEE, 2020. doi:10.1109/ICDE48307.2020.00204.

35 Qingqing Ye, Haibo Hu, Man Ho Au, Xiaofeng Meng, and Xiaokui Xiao. LF-GDPR: A framework for estimating graph metrics with local differential privacy. *IEEE Trans. Knowl. Data Eng.*, 34(10):4905–4920, 2022. doi:10.1109/TKDE.2020.3047124.

36 Hailong Zhang, Sufian Latif, Raef Bassily, and Atanas Rountev. Differentially-private control-flow node coverage for software usage analysis. In *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, pages 1021–1038. USENIX Association, 2020. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/zhang-hailong.

37 Xiao Zhou and Takao Nishizeki. Graph coloring algorithms. *IEICE Transactions on Information and Systems*, 83(3):407–417, 2000.