Edge Local Differential Privacy for Graph Statistic

1. Introduction:

1.1 Background and Motivation

In today's world, using large amounts of data to make decisions is very important for businesses. Companies use data to improve their services, understand their customers, and stay ahead of their competitors. However, with more data being collected, there are also more worries about keeping this data private and secure. It's a big challenge to make sure that personal information is protected while still getting useful insights from the data.

To deal with these worries, researchers have created different methods to keep data private. One popular method is called differential privacy. This method makes sure that the results of data analysis do not reveal any personal information about individuals in the dataset. This project focuses on a special type of differential privacy called local differential privacy (LDP). LDP allows data to be made private before it is shared with anyone else.

In this report, I extend the work of two studies [1] and [2] to create a cryptography-assisted method to help count the number of k-stars in a graph while keeping the data private. A k-star is a key structure in graph theory, where one central node is connected to k other nodes. Counting k-stars is useful in many areas, like analyzing social networks and studying biological data.

1.2 Objective of the Project

The main goal of this project is to design and implement a method for publishing the number of k-stars in a graph while preserving the privacy of the data. By extending the techniques proposed in [1] and [2], I incorporate cryptographic methods to enhance the privacy guarantees provided by local differential privacy. This approach aims to balance the trade-off between data utility and privacy, ensuring that the published statistics are both accurate and secure.

1.3 Structure of the Report

This report is structured as follows:

- Section 2 provides an overview of the importance of privacy for businesses, particularly in the context of the General Data Protection Regulation (GDPR).
- Section 3 introduces the concept of differential privacy and its relevance, with a focus on local differential privacy.

- Section 4 details the algorithm of my proposed cryptography-assisted method for counting k-stars in a graph.
- Section 5 presents the evaluation and results of my method, including experimental setup, results, and comparison with existing methods.
- Section 6 discusses the limitations of my approach.
- Section 7 concludes the report with a summary of my findings, suggestions for future work, and final thoughts.

2. Importance of Privacy for Business:

Privacy is becoming more important for businesses because of the new law, the General Data Protection Regulation (GDPR), enacted by the European Union. GDPR has made privacy a top priority for businesses that handle the personal data of EU citizens, regardless of where the business is located.

The GDPR sets strict guidelines on how personal data should be collected, stored, and used. It aims to give individuals more control over their personal information and ensure that businesses handle this data responsibly. Key principles of GDPR include [5]:

- **Lawfulness, Fairness, and Transparency**: Data must be processed legally and openly, with organizations required to inform individuals about how their data is being used.
- **Purpose Limitation**: Data should be collected for specific, legitimate purposes and not used in ways incompatible with those purposes.
- **Data Minimization**: Only the data necessary for the specified purposes should be collected and processed.
- **Accuracy**: Data must be kept accurate and up-to-date.
- **Storage Limitation**: Data should not be retained longer than necessary for the purposes for which it was collected.
- **Integrity and Confidentiality**: Data must be processed securely to protect against unauthorized access, alteration, or loss.
- **Accountability**: The data controller is responsible for being able to demonstrate GDPR compliance with all of these principles.

Even though the regulation is only enforceable within the European Union, its impact is global. Companies outside the EU that handle data of EU citizens must also comply with GDPR standards. This extraterritorial reach ensures that multinational corporations adhere to stringent data protection practices, irrespective of their geographic location.

3. Preliminaries:

In this section, we explain the concept of differential privacy (DP) in the context of graph data. We will explore the basic principles of DP, local differential privacy (LDP), and how these techniques are applied to graphs.

3.1 Introduction to Differential Privacy

Differential privacy (DP) is a mathematical framework designed to provide strong privacy guarantees for individuals in a dataset. It ensures that the output of any analysis is insensitive to any single individual's data, making it difficult for an observer to determine whether any specific individual is included in the dataset.

The key idea behind DP is to add controlled noise to the data or the query results. The privacy loss parameter, ε, quantifies the privacy guarantee: smaller values of ε provide stronger privacy but introduce more noise, reducing the accuracy of the results. This trade-off between privacy and accuracy is a fundamental aspect of DP.

Formally, for an algorithm A and any two datasets D_1 and D_2 that differ by at most one element, DP ensures that:

$$\mathrm{P}(\mathrm{A}(\mathrm{D1}) \in \mathrm{S}) \leq \mathrm{e}^{\epsilon} \cdot \mathrm{P}(\mathrm{A}(\mathrm{D}_2) \in \mathrm{S})$$

for all possible outputs S.

3.2 Local Differential Privacy (LDP)

Local Differential Privacy (LDP) is a variant of DP where data is randomized on the user's side before being sent to the data collector. This approach does not require the user to trust the data collector, as the data is privatized at the source.
In LDP, each user's data is perturbed independently using a privacy-preserving mechanism. The data collector aggregates these noisy responses to perform analysis. LDP is particularly useful in scenarios where raw data centralization poses significant privacy risks.

For instance, in a social network, instead of collecting the exact number of friends each user has, LDP techniques might involve users adding random noise to their friend counts before reporting. This way, the social network can estimate the overall distribution of friend counts without knowing the exact number for any individual user.

3.3 Differential Privacy in Graphs

Applying differential privacy to graphs involves specific challenges due to the nature of graph data. Graphs consist of nodes (vertices) and edges (connections between nodes), and privacy can be defined at the node or edge level.

Edge Differential Privacy (Edge DP): Protects the presence or absence of individual edges. Two graphs are considered neighbors if they differ by a single edge. This means an adversary cannot determine whether a specific connection between two nodes exists.

Node Differential Privacy (Node DP): Provides stronger privacy by protecting the presence or absence of nodes, including all their associated edges. Two graphs are neighbors if they differ by one node and all edges connected to that node. Node DP is more challenging because changing one node can affect multiple edges.

3.4 Sensitivity in Differential Privacy

Sensitivity measures the impact of a single change in the dataset on the output of a function. In the context of graphs, sensitivity can be high due to nodes with many connections (high degree).

Global Sensitivity: Considers the maximum change in the output of a function across all possible pairs of neighboring datasets. For graphs, this can be too pessimistic, especially in networks with nodes of very high degree.

Local Sensitivity: Focuses on the sensitivity of a function at a particular dataset, considering only the actual neighbors of the dataset. This can provide a more practical measure for adding noise.

Graph Projection: To manage high sensitivity in graphs, techniques like graph projection are used to bound the maximum degree of nodes, reducing the overall sensitivity and making differential privacy more feasible.

3.5 k-star definition

A k-star in a graph is a structure where a central node is connected to k other nodes. Counting k-stars is a common task in graph analysis, providing insights into the local structure of the network. However, this task can reveal sensitive information about the nodes and edges in the graph. To address this, we use LDP to add noise to the data at the user's end before it is aggregated, ensuring that individual data points remain private.

3.6 Laplace mechanism

To achieve differential privacy, a common approach is to add Laplace noise on it. Suppose we want to achieve ε-DP with the function with sensitivity s, then adding noise following Laplace distributed centered at 0 with scale (s/ε) to the data can do the work.

With these preliminaries, my project extends previous work to develop a cryptography-assisted method for counting k-stars in a graph under local

differential privacy. This approach ensures strong privacy guarantees while allowing accurate analysis of network structures.

3.7 l2 loss

We will use l2 loss (i.e., squared error) for utility metric. To calculate l2 loss between two number a, b, the l2 loss is defined as follow:

$$l_2(a, b) = (a - b)^2$$

The reason for choosing l2 loss is because [1] also use l2 loss as their metric, and it is a simple metric for evaluating the difference before and after applying our privacy algorithm.

4. Proposed method:

In this section, we describe the proposed cryptography-assisted method for counting k-stars in a graph while ensuring local differential privacy (LDP). My approach builds on previous work and incorporates cryptographic techniques to enhance privacy guarantees. The primary goal is to enable accurate counting of k-stars without compromising the privacy of individual data points.

4.1 Previous work

In [1], J. Imola et al. proposed method for publishing the number of k-star in a graph with local differential privacy. The algorithm is as follow:

Data: Graph G represented as neighbor lists $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \{0,1\}^n$, privacy budget $\varepsilon \in \mathbb{R}_{\geq 0}$, $\tilde{d}_{max} \in \mathbb{Z}_{\geq 0}$.
Result: Private estimate of $f_{k\star}(G)$.

$\Delta \leftarrow \binom{\tilde{d}_{max}}{k-1}$;
for $i = 1$ **to** n **do**
  $\mathbf{a}_i \leftarrow \texttt{GraphProjection}(\mathbf{a}_i, \tilde{d}_{max})$;
  /* d_i is a degree of user v_i. */
  $d_i \leftarrow \sum_{j=1}^{n} a_{i,j}$;
  $r_i \leftarrow \binom{d_i}{k}$;
  $\hat{r}_i \leftarrow r_i + \mathrm{Lap}\left(\frac{\Delta}{\varepsilon}\right)$;
  $release(\hat{r}_i)$;
end
return $\sum_{i=1}^{n} \hat{r}_i$

Algorithm 1: LocalLap$_{k\star}$

In the algorithm, a graph is projected to graph with lower maximum degree, then each node computes the number of k-stars centered at it. Since the maximum degree is bounded, the sensitivity can be computed using the maximum degree instead of the number of nodes. Since adding an edge will increase at most $\binom{\tilde{d}_{max}}{k-1}$

k-stars, therefore the sensitivity is $\binom{\tilde{d}_{max}}{k-1}$. The algorithm then publishes the number with Laplace noise.

Two problems occur when using this algorithm: First, how do we decide the maximum degree? The maximum degree is important because it directly influences the sensitivity of the k-star counting function. Higher maximum degrees result in higher sensitivity, requiring more noise to ensure differential privacy, which can degrade the accuracy of the results. Lowering the maximum degree reduces sensitivity and the amount of noise needed, improving accuracy. However, setting the maximum degree too low leads to significant information loss because the graph no longer accurately reflects the original network structure. Important structural details can be lost, impacting the integrity of the graph analysis.

Second, how do we project the graph to lower degree? To project a graph to lower degree, it's essential that some information of the graph will be discard, then what is the best way to do it becomes a problem for this algorithm.

In [2], S. Liu et al. introduce a solution for choosing the parameter for the maximum degree used for publishing degree distribution histogram.

Algorithm 3 Crypto-assisted parameter selection

Input: Adjacent bit vectors $\{B_1, ...B_n\}$,
security parameters a, b
Output: Projection parameter θ

for each integer $k \in \{1, 2, ..., K\}$ **do**
  /* User side. */
  for each user $i \in \{1, 2, ..., n\}$ **do**
    $\hat{d}_i \leftarrow$ `LocalProjection`(B_i, k) // Sec. V
    $d_i \leftarrow \sum_{j=1}^{n} b_{i,j}$
    $noise \leftarrow randint(0, a-1)$
    r $\leftarrow$ `PRG`$(seed)$
    $mask = \sum_{j=i+1}^{n-1} r_{i,j} - \sum_{j=1}^{i-1} r_{i,j}$
    $Enc_{T_{k,i}} \leftarrow a * |d_i - \hat{d}_i| + b + noise + mask$
    User i sends $Enc_{T_{k,i}}$ to server
  end for
  /* Curator side. */
  $Enc_{T_k} \leftarrow \sum_{i=1}^{n} Enc_{T_{k,i}}$
  $\theta \leftarrow k$ when $(Enc_{T_k} + E_D)$ is minimum
end for
return θ

The key idea for their algorithm is to use order preserving encryption to encrypt the projection loss calculated by each node after projection. Then, aggregate the result, and choose the parameter that produce the lowest utility loss + projection loss. The paper use a linear order preserving scheme, and add a mask to perform secure aggregation for all nodes.

The problem when using this is that how do we calculate the utility loss and projection loss in the graph? Although we know there will be utility loss and projection loss after the projection, modeling the amount of loss is another problem if we want to use this scheme.

As for how to project a graph to lower degree, in [3] Wei-Yen Day et al. proposed a method.

Algorithm 1 π_θ: projection by edge-addition.

Input: An input graph $G = (V, E)$, a degree bound θ, and a stable edge ordering $\Lambda = \langle e_1, \cdots e_n \rangle$.
Output: An output θ-bounded graph $\pi_\theta(G)$.

$E^\theta \leftarrow \emptyset$; $d(v) \leftarrow 0$ for each $v \in V$
for $e = (u, v) \in \Lambda$ **do**
  if $d(u) < \theta$ & $d(v) < \theta$ **then**
    $E^\theta \leftarrow E^\theta \cup \{e\}$; $d(u) \leftarrow d(u) + 1$; $d(v) \leftarrow d(v) + 1$
  end if
end for
return $G^\theta = (V, E^\theta)$

The method requires stable edge ordering, that is, the edge have a kind of consistent ordering. This can be achieved by giving node number, and order the edge by the node. The algorithm is to add the edge in the order one at a time. When adding an edge, we check if the nodes of the edges have degree less than the maximum degree. If not, we do not add this edge.

In [1], J. Imola et al. also proposed a method for projecting the graph. The projection is simple: if the node's degree is higher than the maximum degree, set the degree to the maximum degree for that node. This method

Extending these works, I build up a scheme to publish the number of k-stars in a graph with edge local differential privacy.

4.2 Proposed Method

In my proposed method to calculate the number of k-star, I use the same algorithm from [1]. To decide the maximum degree of the graph, I use the idea of [2] to make the decision. Also, I compared the projection method proposed in [1] and [3] to see the result.

To apply the cryptography assisted method in [2] to my method, we need another way for estimating the utility loss and projection loss. In [2], the projection loss is estimated by calculating how many edges are removed, and the utility loss is calculated by the expected value of loss produced by Laplace noise. However, if we apply the same estimation directly when calculating the projection loss of k-star, a problem occurs is that it cannot reflect the projection loss well. The projection loss is often overwhelmed by utility loss if calculating the number of removed edge as projection loss directly, thus producing small maximum degree as output.

To get around with this situation, we need to re-evaluate the projection loss so that it's comparable to the utility loss. The calculation of utility loss is by using the variance of Laplace noise, which can be viewed as the expected l2 loss generated by Laplace noise. Therefore, the projection loss should also be a l2 loss generated by the removal of the edges. Continuing the thought process, we know that changing the degree of a node from d_1 to d_2 will cause the number of k-star changed from $\binom{d_1}{k}$ to $\binom{d_2}{k}$. Therefore, we can re-define the projection loss to be the l2 loss between $\binom{d_1}{k}$ and $\binom{d_2}{k}$.

5. Evaluation and Results:

5.1 Experimental Setup

For the evaluation of my method, we used the SNAP dataset from Stanford [4], specifically the Facebook dataset. The Facebook dataset contains anonymized social network data, which is ideal for testing my privacy-preserving k-star counting algorithm. Also, we use artificial dataset generated using Erdős-Rényi graph model.

The Facebook dataset includes nodes representing users and edges representing friendships between them. This dataset is particularly suitable due to its rich structure and the presence of both high-degree and low-degree nodes, allowing us to thoroughly evaluate the performance of my method under different conditions.

Erdős-Rényi graph, also known as a binomial graph, takes 2 parameters as input: number of nodes and the probability for generating each edge. It is a simple model we can use for generating graph with desired size for our experiment.

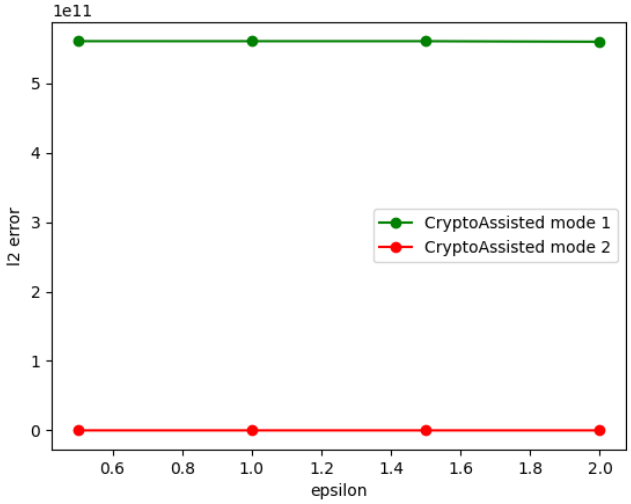
To assess the utility of my approach, we use the l2 loss as described in section **3.7**.

I compare the result after applying parameter selection method of mine and method in [1]. Also, I compare the result of projection method in [1] vs method in [3].

5.2 Results and Analysis

CryptoAssisted Method Compare

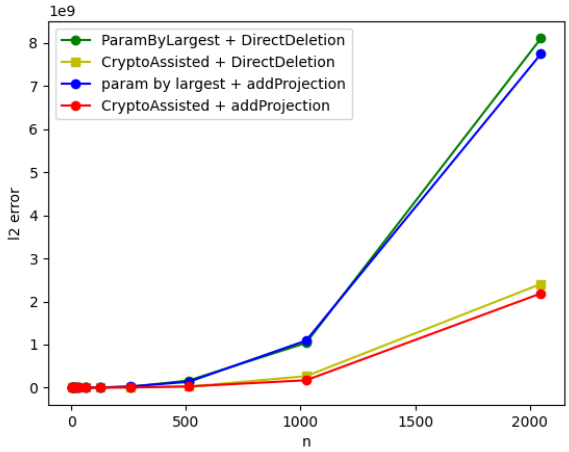
First, we show that my method actually performs better compare to the un-modified method in [2]. In the figure, mode 1 represents the original method in [2], and mode 2 represents my method. We can see that there is a huge difference in l2 loss between two methods. My method out-performs the original method in this use case by a lot.



There are four types of set up in the experiment: ParamByLargest + DirectDeletion, CryptoAssisted + DirectDeletion, ParamByLargest + AddProjection and CryptoAssisted + AddProjection. **ParamByLargest** means that the choose of maximum degree is by publishing degree of each node with Laplace noise and choose the maximum. **CryptoAssisted** is my method described in **4.2. DirectDeletion** means the projection is done by directly setting maximum degree as degree number for nodes that have higher degree than the maximum degree, and **AddProjection** uses the method proposed in [3] that add an edge each time if the nodes have lower degree.

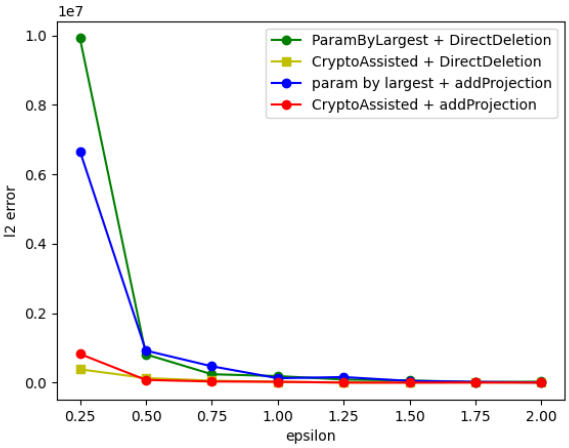
Relation between n and l2 loss.

We evaluate the l2 loss with the four methods while changing node number n. The result shows that l2 loss grows when node number increases, and cryptography assisted method performs better than estimating by largest degree method. However, two projection methods look similar to each other, and do not have noticeable difference in terms of l2 loss.



Relation between ϵ and l2 loss.

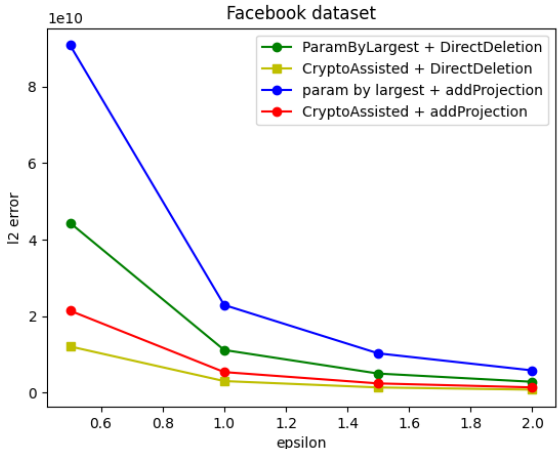
Next, we evaluate the l2 loss with different privacy budget ϵ. The privacy budget is divided into 2 equal budgets $\epsilon/2$ for projection and estimating parameter in the cases of non-crypto selection tasks. The result also shows that with cryptography assisted method, we can lower the l2 loss by a lot with the same privacy budget. The difference of l2 loss becomes small as privacy budget increases, however, more budget indicates less privacy guarantee, and cryptography assisted method can gain more value when trying to give more privacy guarantee.



Performance on Facebook dataset with different ϵ.

Finally, we apply our method on real dataset. The Facebook dataset have 4038 nodes and 88234 edges in it. As we can see, our crypto assisted method has significant advantages at low privacy budget, and can still perform better with higher privacy budget. Add projection method seems to perform worse in this dataset. This is reasonable because add projection although saves the structure of the graph, but when counting the number of k-star, the only thing matter is the degree of each node, and directly setting degree to the maximum degree will make less k-star disappear when counting them. Therefore, though add projection

provides more general usage when processing graph, direct deletion method is more suitable for this case.



6. Discussion:

6.1 Time Complexity

The time complexity of the proposed parameter choosing method is $O(n^2)$ for each node since the secure aggregation takes $O(n)$, and there are n possible maximum degree. As comparison, the time complexity of method in [1] is $O(1)$ since each node only have add noise to their degree without any other operation.

As for projection method, the time complexity of method in [1] is also $O(1)$ since each node only have to set its degree to the maximum degree if it has larger degree. The add projection by [3] requires $O(|E|)$ complexity since it needs to run through all the edges. In the experiment, we see that these two methods have similar performance, neither of them out performs the other. Therefore, project the graph by directly setting the degree to maximum degree would be good for this use case.

6.2 Limitations

As for the limitations, there are mainly two limitations. **First, the time complexity**: The time complexity of my method is $O(n^2)$, which can be prohibitive for very large graphs. In contrast, the method proposed in [1] has a time complexity of $O(1)$ for each node, making it more scalable for large-scale applications. The quadratic time complexity of our approach means that as the number of nodes n increases, the computational resources required grow significantly, potentially limiting its practical applicability for massive datasets.

Second, the method's specificity: Our method is specifically designed for counting the number of k-stars in a graph. While it performs well for this particular graph statistic, it is ad-hoc and may not generalize easily to other types of graph

statistics. Applying this method to different graph statistics would require modifications to the details of estimating utility loss and projection loss. Each new application would necessitate a tailored approach to ensure that privacy is preserved without compromising the accuracy and utility of the statistic being measured.

These limitations highlight the need for further research to improve the scalability and generality of our method. Enhancing the time complexity would make it more suitable for larger graphs, while developing a more flexible framework could extend its applicability to a broader range of graph statistics.

7. Conclusion:

7.1 Summary of Finding

In this report, we introduced a cryptography-assisted method for counting the number of k-stars in a graph while ensuring local differential privacy (LDP). Our approach extends previous work by incorporating cryptographic techniques to enhance privacy guarantees and employs a mechanism for graph projection to manage sensitivity.

Key findings include:

1. **Effectiveness of Cryptography-Assisted Method:**
   - Our cryptography-assisted method significantly improves the privacy-utility trade-off compared to previous approaches. The use of cryptographic techniques ensures that sensitive information remains protected while allowing accurate counting of k-stars.
2. **Evaluation on the Facebook Dataset:**
   - Using the Facebook dataset from the SNAP collection, we demonstrated the practicality of our method. The l2 error metric was employed to evaluate the utility of our approach, with results indicating that the method effectively balances privacy and accuracy.
3. **Limitations**:
   - Our method has a time complexity of $O(n^2)$, which may limit its scalability for very large graphs. Additionally, it is specifically designed for counting k-stars and may require significant modifications to be applied to other graph statistics.

7.2 Future Work

While our method for privacy-preserving k-star counting shows promising results, there are several avenues for future work that could further enhance its

applicability, and generalizability. Here are some ideas for future research and development:

1. **Improving Scalability:**
    - Optimizing Algorithms: Investigate more efficient algorithms to reduce the time complexity from $O(n^2)$ to potentially linear or near-linear time complexity. This could involve leveraging advanced data structures or more sophisticated optimization techniques to handle large datasets more efficiently.
2. **Generalizing to Other Graph Statistics**:
    - **Extending Methodology**: Adapt the current method to apply to other graph statistics, such as counting triangles, computing centrality measures, or detecting communities. This would involve developing new techniques for estimating utility loss and projection loss specific to each statistic.
    - **Utility-Privacy Trade-off Framework**: Create a more flexible framework that can balance utility and privacy for a variety of graph statistics, enabling broader application in different types of graph analysis.
3. **Enhanced Empirical Evaluation**:
    - **Benchmarking on Various Networks**: Evaluate the method on a wider range of real-world datasets, including social networks, biological networks, and communication networks, to understand its performance and limitations across different contexts.
    - **User Studies**: Conduct user studies to assess the practical impact of the method on real-world applications, gathering feedback from practitioners to guide further improvements.
4. **Real-World Applications and Deployment**:
    - **Integration with Privacy Tools**: Work on integrating the method into existing privacy-preserving data analysis tools and platforms, making it accessible to a broader audience.
    - **Case Studies**: Collaborate with industry partners to apply the method in real-world scenarios, such as social network analysis, recommendation systems, and cybersecurity, to demonstrate its practical benefits and challenges.

By addressing these areas, future work can significantly advance the state of privacy-preserving graph analysis, making it more efficient, versatile, and applicable to a wide range of real-world problems.

8. References:

[1] J. Imola, T. Murakami, and K. Chaudhuri, “Locally differentially private analysis of graph statistics,” in 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 983–1000.

[2] S. Liu, Y. Cao, T. Murakami and M. Yoshikawa, "A Crypto-Assisted Approach for Publishing Graph Statistics with Node Local Differential Privacy," *2022 IEEE International Conference on Big Data (Big Data)*, Osaka, Japan, 2022, pp. 5765-5774,
doi: 10.1109/BigData55660.2022.10020435

[3] Wei-Yen Day, Ninghui Li, and Min Lyu. 2016. Publishing graph degree distribution with node differential privacy. In Proceedings of the 2016 International Conference on Management of Data. 123–138.

[4] Leskovec, J., & Krevl, A. (2014). SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. http://snap.stanford.edu/data

[5] What is GDPR, the EU’s new data protection law? (gdprinfo.eu)