Privacy-Preserving Triangle Counting in Large Graphs

Xiaofeng Ding*, Xiaodong Zhang*, Zhifeng Bao†, Hai Jin*
*Services Computing Technology and System Lab
Cluster and Grid Computing Lab
School of Computer Science and Technology
Huazhong University of Science and Technology, P. R. China
†RMIT University, Australia
{xfding,sara7,hjin}@hust.edu.cn,zhifeng.bao@rmit.edu.au

ABSTRACT

Triangle count is a critical parameter in mining relationships among people in social networks. However, directly publishing the findings obtained from triangle counts may bring potential privacy concern, which raises great challenges and opportunities for privacy-preserving triangle counting. In this paper, we choose to use differential privacy to protect triangle counting for large scale graphs. To reduce the large sensitivity caused in large graphs, we propose a novel graph projection method that can be used to obtain an upper bound for sensitivity in different distributions. In particular, we publish the triangle counts satisfying the node-differential privacy with two kinds of histograms: the triangle count distribution and the cumulative distribution. Moreover, we extend the research on privacy preserving triangle counting to one of its applications, the local clustering coefficient. Experimental results show that the cumulative distribution can fit the real statistical information better, and our proposed mechanism has achieved better accuracy for triangle counts while maintaining the requirement of differential privacy.

CCS CONCEPTS

• **Security and privacy** → Data anonymization and sanitization;

KEYWORDS

Differential privacy; Triangle counting; Large graph

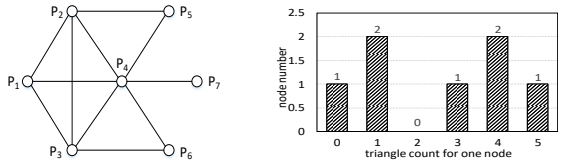
ACM Reference Format:
Xiaofeng Ding, Xiaodong Zhang, Zhifeng Bao, Hai Jin. 2018. Privacy Preserving Triangle Counting in Large Graphs. In 2018 ACM Conference on Information and Knowledge Management (CIKM'18), October 22–26, 2018, Torino, Italy. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3269206.3271736

1 INTRODUCTION

Graph data are key parts of big data and widely used for modelling complex structured data with a broad spectrum of applications. For example, the explosion of social network in the last decade leads to very large scale graph data. In this paper, we focus on the triangle counting in graph data [8, 9]. Essentially, in a graph G, a triangle is a set of three nodes among which each pair of nodes are neighbours in G. Ever since Watts et al. [24] introduced the measurement to determine whether a graph is a small-world network, many work focus on finding the triangles, which are one of the simplest but effective descriptions of a node's status in a small-world [7, 16, 25].

However, as shown in Example 1.1, users' privacy in social networks can be leaked through the triangle count distribution. In fact, attacking from neighbours (e.g., friends in a social graph) has been proven to be feasible [22]. Triangles linked to one node insinuate the relationships around this individual, and thereby it can easily disclose his/her status in the network.

Example 1.1. Given a social network (shown in Fig. 1(a)) which contains seven users from P_1 to P_7. To publish the statistical result about the triangle count distribution, we need to publish a histogram like Fig. 1(b) to the public. In other words, this histogram contains the answer of the query: *How many nodes in the graph that each of them connects to x triangles?* Here, x is the triangle counts of each node in the graph. Although the labels of all individuals have been removed, such histogram can still reveal users' privacy when publishing its triangle count distribution. Suppose there is an attacker knowing everyone in this network except P_4, and s/he is not sure whether P_4 (say Alice) is the person s/he knows in the real world. Once s/he gets the triangle count distribution of this network that we published on the Internet, then by comparing the knowledge of all users s/he has confirmed in this graph, s/he can find there is an unknown person (P_4) who connects with 5 triangles. That is to say, this unknown person is the friend of at least four other people in this network. As this attacker has known in real world that Alice is the common friend of most users in this graph, the attacker can easily infer that P_4 is Alice.



(a) An example of input graph (b) Distribution of Triangle Count

Figure 1: Motivating Example

Therefore, in this paper we study how to preserve users' private information leaked from their triangle counts. In general, privacy

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3271736

protection can be achieved by two types of methods: anonymization based protection and perturbation based protection. The anonymization based methods are mainly designed for ad hoc attacks like structure attack, which is vulnerable for strong adversaries. However, differential privacy, as a representative standard in perturbation based approaches, makes the strongest assumption in protecting privacy from graph data. Moreover, we observe that differential privacy is widely adopted in the protection for statistical features, which is totally different from graph anonymity. Our contributions can be summarized as follows:

- *W*e propose and study the problem of privacy preserving triangle counts of each node for the first time. Unlike previous work using edge-differential privacy as principles, we manage to preserve triangle counts with more strict definition on node-differential privacy. (Section 3)
- *W*e propose a novel projection method to make the original graph more useful when satisfying node-differential privacy. We prove that when publishing a triangle count distribution with threshold λ, an upper-boundary $4\lambda + 1$ of global sensitivity can be achieved. (Section 4)
- *W*e extend our method to the local clustering coefficient and publish these coefficients by dividing them into k groups. We also prove that the sensitivity upper-boundaries are $4\lambda + 1$ for coefficient distribution and $2\lambda(k-1) + k$ for cumulative coefficient distribution. (Section 5)
- *W*e conduct comprehensive experiments over three large real-world datasets, and the results demonstrate that our scheme is very effective for triangle counts protection. (Section 6)

2 RELATED WORK

Due to the popularity of graph data, many recent work have been proposed to study the privacy preserving graph data publishing. As mentioned in Section 1, these preserving methods can be divided into two groups: anonymization based protection and perturbation based protection, while the former focuses on publishing similar graphs which cannot be distinguished from each other, and the differential privacy is used more often when users want to publish statistical features of graphs.

K-anonymity was first proposed by Sweeney [21], but it cannot defend the homogeneity attack and background knowledge attack. Then many methods were proposed to overcome such drawbacks such as l-diversity [18] and t-closeness [15], which can protect both identifications and relationships in data. In addition, many approaches based on k-anonymity were proposed in recent years, including k-NA (k-Neighborhood Anonymity) [30], k-DA (k-Degree Anonymity) [17], and k-auto (k-automorphism) [1], which protect graph privacy from the angles of node neighbor, node degree, and subgraph isomorphic.

However, none of the above models can resist the background attacks, and this promotes the development of methods based on differential privacy, which has been applied in various fields [14, 26]. The concept of differential privacy was first proposed by Dwork et al. [5], which relies on certain statistic values of the relevant data to protect individual privacy. Differential privacy can be described as this: if we have two datasets differ only by one record, we also hold the strongest background knowledge of this database; however, we cannot track this record from any query result. Dwork et al. [5] proved that adding Laplace noise to the output can satisfy differential privacy. After that McSherry proposed an exponential mechanism [19] that could fulfill any type of output, but it comes with the disadvantage of high algorithm complexity. Then it has been observed that when combining the differential privacy with graphs, it can be a better way to publish the results of data mining than to publish the data itself. Thereby noises are added to the histogram to result in distribution histograms that satisfy the differential privacy [27, 29]. Generally, publishing graphs based on differential privacy can be roughly divided into two categories: edge-differential privacy [10, 11, 28] and node-differential privacy [2, 3, 12]. In this paper, we choose the node-differential privacy to protect the important factors when publishing the results of triangle counts in graph.

Triangle counting is a well-recognized important research topic on graphs; however, as far as we are concerned, no previous work has focused on privacy preserving on each node's triangle counting, which is a combined query between triangle counts and node number. Do and Ng [4] had committed to solve the problem of preserving some short cycles of a graph such as triangles in a distributed environment by modifying an existing secure matrix computation protocol. Their work focused on optimizing the encryption matrix to achieve multi-party secure triangle data transmission, which falls into the category of cryptography. In addition some researches ever focused on single-valued queries, for example, subgraph counting queries such as the number of triangles, or the number of k-stars in a graph. The work in [2] studied the problem of proposing a new metric other than the global or local sensitivity for node-differential privacy to measure the least magnitude of added noises when answering the queries of subgraph counting. Kasiviswanathan et al. [12] extend their proposed LP-based function to release the small subgraph counts, and they worked on the special case of the subgraph who holds three nodes, ie., is the triangle or 2-stars, which is also a single-valued query. Task et al. [23] have ever took the combined query of triangle counting into consideration and they proposed the idea of preserving the local clustering coefficients for nodes in triangulation. To make the global sensitivity small enough, they proposed the concept of outlink-privacy, which is less protective than differential privacy. In this paper, we provide a novel data publishing method, which can not only satisfy the definition of node-differential privacy, but can also solve the problem of large global sensitivity caused in the processing of triangle counting.

3 PRELIMINARIES AND PROBLEM FORMULATION

3.1 Differential Privacy

Differential Privacy [5] is a mathematical guarantee of privacy that satisfies fully privatized queries, which can be defined as below.

Definition 3.1. (Differential Privacy) *A* random query $Q : D \longrightarrow \mathbb{R}^k$ satisfies ϵ-differential privacy (ϵ-DP), if for any two neighboring datasets D_1 and D_2, any possible result R satisfies:

$$\frac{Pr[Q(D_1) = R]}{Pr[Q(D_2) = R]} \leq e^{\epsilon}$$

where ϵ is a parameter for privacy level of query results.

For two neighboring datasets D_1 and D_2, the global sensitivity is defined to represent the largest difference between them.

Definition 3.2. (Global Sensitivity) The global sensitivity of query $f : D \longrightarrow \mathbb{R}^k$ is:

$$\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|_1$$

Laplace Mechanism. To satisfy differential privacy noise is added to the output before publishing. In the Laplace mechanism [6], when we publish a k-ary function $f(D)$ and make it satisfy ϵ-DP, we can use the following formula:

$$Q(D) = f(D) + Lap(\Delta f/\epsilon)^k$$

3.2 Privacy Preservation on Triangle Count

A triangle in a social network means that three nodes are all mutual friends. If there are many triangles appearing from just few nodes, such tight relationship means that they are in a small and nearly independent group. When we describe or compare two groups, triangle count is an essential parameter. Although there are also some other parameters such as counts of star subgraphs or square subgraphs [30] that can be used to compare two graphs, they are all derivatives of triangle counts which have similar properties.

Problem Statement. Given an input graph $G = (V, E)$ with no label, we release the distribution of triangle count on each node, which can be described as the answer of the query: *How many nodes are there in the graph such that each of them connects to x triangles?* Here, x is the triangle counts on nodes in the graph. For privacy preserving purpose, no one can re-identify an individual's private information from the released distribution, and the published distribution should be able to hold similar statistical features as the real one. A formal definition is as below.

Definition 3.3. (Differential Private Triangle Counting) Given a graph G with m nodes, and $R = \{r_1, r_2, r_3, ..., r_m\}$ represents the results of triangle counting for each node in G. Let $R' = \{r'_1, r'_2, r'_3, ..., r'_k\}$ be a variation of R with no repetition (duplicate counts), and was arranged in ascending order. Let $N = \{n_1, n_2, n_3, ..., n_k\}$ be the result of query Q: How many nodes in the graph that each of them connects to R' triangles? (For example, in graph G there are n_1 nodes that each of them has a triangle count of r'_1.) For any two graphs G and G' differ by only one node, the differential private mechanism should make any possible result N satisfies:

$$\frac{Pr[Q(G) = N]}{Pr[Q(G') = N]} \leq e^{\epsilon}$$

For this problem we choose node-differential privacy other than edge-differential privacy to protect the triangle counts for two reasons. (1) In a social network, a node represents an individual in real world and is more significant than an edge. When we talk about privacy, we tend to talk about a person's privacy. Exposing one person himself would be more serious than exposing a relationship of him. (2) Guaranteeing node-differential privacy on triangle counts is much more challenging than edge-differential privacy. In particular, the sensitivity of node-differential privacy on triangle counts is quite high. Suppose we have a graph of n nodes, if we add one node to it, in the worst case, this virtual node is the friend of all users in this network, then it will cause the difference of C_n^2 triangles to be added to the graph. The difference is in an exponential magnitude, which is too large to make the noise uncontrollable. Since the sensitivity is related to the size of the graph, it is impossible to compute an upper-boundary of global sensitivity for triangle counts in the large graph.

3.3 Utility Metrics

Similar to the work [3] of differential privacy preserving degree on graphs, we use L_1 distance to measure the difference between the original triangle count histogram and the noisy one. The L_1 distance can be defined as below.

Definition 3.4. (L_1 Distance) Given two histograms h and h' in a V-dimensional vector space, the L_1 distance between them is

$$\| h - h' \|_1 = \Sigma_{i=0}^{V-1} \mid h_i - h'_i \mid$$

where h and h' are vectors of

$$h = (h_0, h_1, h_2, ..., h_{V-1})$$
$$h' = (h'_0, h'_1, h'_2, ..., h'_{V-1})$$

We also use the KS-distance (two-sample Kolmogorov-Smirnov test) to evaluate our approach of releasing the cumulative histogram, and the KS-distance between two histograms can be explained as below.

Definition 3.5. (KS Distance) Given two histograms h and h', the KS-distance between them is

$$KS(h, h') = \max_i |CDF_{h(i)} - CDF_{h'(i)}|$$

where $CDF_{h(i)}$ represents the value of cumulative distribution of h(i).

4 OUR ALGORITHM

We first present the algorithm and provide the upper-boundary of global sensitivity when releasing the triangle count histogram in Section 4.1. Then in Section 4.2 we show cumulative triangle count histogram together with its sensitivity, and the utility metrics for histograms will be also introduced.

4.1 Edge-deletion Based Data Projection

We propose a novel projection method called T_λ (shown in Alg. 1) by using edge-deletion in the original graph. In this mechanism some extra edges were deleted and we obtain a new graph G^λ. All nodes in G^λ must have their triangle counts no larger than λ.

In Alg. 1 the graph G was put in by an order of node pairs, and this ordering should be stable. We consider a sequence of node pairs is stable iff when we delete a node from the graph, the new sequence is almost the same as before (Definition 4.1).

Definition 4.1. (Stable Sequence) Given two graphs $G = (V, E)$ and $G' = (V', E')$ that differ by only one node, any three nodes v_i, v_j, and v_z in G correspond to v'_i, v'_j, and v'_z in G', a stable node pairs sequence satisfies:

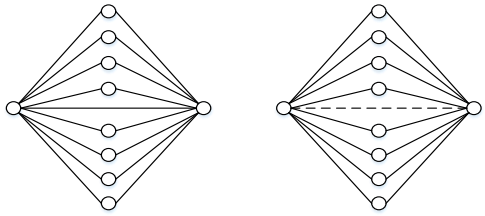
$$v_i\text{-}v_j < v_j\text{-}v_z \Rightarrow v'_i\text{-}v'_j < v'_j\text{-}v'_z$$

Here, v_i-$v_j < v_j$-v_z means in the sequence v_i-v_j is in the front of v_j-v_z.

Algorithm 1 T_λ: projection by edge-deletion

Input: Graph $G = (V, E)$ and a boundary λ for number of triangles of each node
Output: λ-bounded Graph $T_\lambda(G)$
1: $Deg(i) \leftarrow 0$ for all nodes in G
2: **for** $v_i \in V$ **do**
3: $Deg_i(G) \leftarrow$ Query: How many nodes v_i links to?
4: $Tri_i(G) \leftarrow$ Query: How many triangles v_i links to?
5: $LinkNode(v_i) \leftarrow$ sequence of nodes link to v_i
6: **for** $v_i \in V$ **do**
7: **while** $Tri_i > \lambda$ **do**
8: $MaxTemp \leftarrow 0, k \leftarrow 0$
9: **for** $v_j \in LinkNode(v_j)$ **do**
10: **if** $Deg_j > MaxTemp$ **then**
11: $MaxTemp \leftarrow Deg_j, k \leftarrow j$
12: $G \leftarrow$ Delete edge v_i-v_k
13: $Tri_i \leftarrow$ Query: How many triangles v_i links to?
14: $Tri(G) \leftarrow$ Update the triangle counts of all nodes in G
return $G^\lambda = (V, E^\lambda)$

$T_\lambda(G)$ generates a graph whose triangle counting for each node is bounded by λ. To make G^λ save more triangles, we choose to delete the edges that link their neighbor nodes with larger degree (lines 6 to 14). Since deleting one edge might cause all triangles disappear (shown in Fig. 2), which edge should be deleted first becomes an essential problem.



(a) the original graph (b) delete by one edge

Figure 2: Deleting an edge from a special graph may cause all triangles disappear

There are three options for us to choose: (1) DR: delete edges randomly; (2) DL: delete edges from the linked nodes which have larger degree; (3) DS: delete edges from the linked nodes which have smaller degree. For a complete graph these three methods work in the same way. However, most graphs in real life tend to contain some super star nodes and lots of general nodes. Thus, deleting edges randomly is uncontrollable and the results are unpredictable, so we focus on the other two methods. In this paper we use DL to obtain our λ-bounded graph $T_\lambda(G)$, and the performance of DL and DS with different choices of λ will be analyzed in Section 6.1.

Alg. 2 presents the process of how to achieve a noisy triangle count distribution by adding noise to the original histogram. We input a raw graph G and the privacy budget ϵ, and then we use Alg. 1 to set a threshold. Finally we use the Laplace mechanism to preserve the graph and output the histogram.

In our projection, the choice of λ depends on not only the data distribution, but also the privacy parameter ϵ. If we set a larger threshold λ, then we can save more triangles from the original graph, but the noise added to the histogram will be larger. On the contrary, when we set a smaller threshold, the noise will be smaller but less triangles will be preserved. Thereby in experiments we study different λ settings in order to balance the threshold and the noise.

Algorithm 2 Tr^λ-Histogram: triangle count distribution

Input: Graph $G = (V, E)$ and privacy budget ϵ
Output: A noisy triangle count distribution Tr^λ
1: $G^\lambda \leftarrow T_\lambda(G)$ by Alg. 1
2: **for** $v_i \in V$ **do**
3: $h_i(G^\lambda) \leftarrow$ *Query: How many nodes link to i triangles?*
4: **for** $v_i \in V$ **do**
5: $Tr_i^\lambda \leftarrow h_i(G^\lambda) + Lap(\frac{4\lambda+1}{\epsilon})$
return Tr^λ

When we release a histogram satisfying differential privacy, an important parameter needs to be considered is the global sensitivity, which can be represented as Δ_{hist}. We release the triangle count histogram $Tr\text{-}hist(G)$ with the upper-bounded sensitivity $\Delta_{hist\text{-}tri}$ of $4\lambda + 1$.

Lemma 4.2. *Given any two graphs G and G' which differ by only one node, we have*

$$\| Tr\text{-}hist(T_\lambda(G)) - Tr\text{-}hist(T_\lambda(G')) \|_1 \leq 4\lambda + 1$$

where T_λ is the function of Alg. 1.

Proof. Let v' be the node in $G' = (V', E')$ and not in $G = (V, E)$, and we have $V' = \{V, v'\}$. Assume all triangles in G' but not in G have the ordering of $\phi = t_0, t_1, t_2, ..., t_{m-1}$, and ϕ_0 represents triangle t_0 is only in G'. The length of ϕ is m, and it is the number of triangles in sequence ϕ. Clearly, these m triangles have the same node v' altogether and we have $m \leq \lambda$. When we delete node v' from G', the process can be described as the same process for deleting all the triangles in ϕ. Every triangle in ϕ may change the other two nodes' triangle counting results, and the difference for these two nodes is 1. In the worst situation, all the triangles in ϕ do not have the shared node except v', then the number of nodes influenced by triangles in ϕ is at most $2m$. For all nodes that influenced by ϕ, each node will cause a difference of at most 2 in histogram. Considering v' brings another difference of 1 in histogram, we get the difference between G' and G, that is $4m + 1$. As we all know $m \leq \lambda$, therefore, we have $4m + 1 \leq 4\lambda + 1$ and $\Delta_{hist\text{-}tri} \leq 4\lambda + 1$. □

4.2 Cumulative Histogram and Utility Metrics

The above section presents a new method to publish the triangle count distribution with sensitivity $Lap(\frac{4\lambda+1}{\epsilon})$. However, sensitivity is still higher than that we wanted. Another way to make the noise magnitude smaller is to publish a cumulative histogram [3], which is the answer of *Query: How many nodes are there that connect no*

more than x triangles? We prove that this cumulative histogram $TC\text{-}hist(G)$ holds a global sensitivity $\Delta_{hist\text{-}ctri}$ smaller than $2\lambda + 1$.

Lemma 4.3. *Given any two graphs G and G′ differing only by one node, we have*

$$\| TC\text{-}hist(T_\lambda(G)) - TC\text{-}hist(T_\lambda(G')) \|_1 \leq 2\lambda + 1$$

where T_λ is the function in Alg. 1.

Proof. We follow the mathematics symbol used in Lemma 4.2, and assume node v' links to m triangles. We first consider v' itself, when we delete v' from G', all bins in cumulative histogram from m to λ will change 1 and this causes the difference of $\lambda - m + 1$. Then we consider all nodes that linked to v'. Because of the deletion, each triangle connected to v' must influence two other nodes, in the worst situation, deleting v' will influence at most $2m$ different nodes. In the cumulative histogram the difference caused by each of the $2m$ nodes is 1. By adding them all up, finally we get the boundary of global sensitivity $\lambda + 1 + m$, and $m \leq \lambda$, therefore, we get $\Delta_{hist\text{-}ctri} \leq 2\lambda + 1$. □

Alg. 3 shows the process of how to obtain a noisy cumulative triangle count distribution by adding noise to the original cumulative histogram. We input a pending graph G and the privacy budget ϵ, and then we use Alg. 1 to set a threshold for it, just the same as Alg. 2. Before adding noises to the result, we change the histogram to be a cumulative histogram (lines 4 to 5). Finally we use the Laplace mechanism to perturb this histogram.

Algorithm 3 TC^λ-Histogram: cumulative histogram distribution

Input: Graph $G = (V, E)$ and privacy budget ϵ
Output: A noisy cumulative triangle count distribution TC^λ

1: $G^\lambda \leftarrow T_\lambda(G)$ by Alg. 1
2: **for** $v_i \in V$ **do**
3: $\quad h_i(G^\lambda) \leftarrow$ *Query: How many nodes link to i triangles?*
4: **for** $h_i \in hist(G^\lambda)$ and $h_i \leq \lambda$ **do**
5: $\quad h_i \leftarrow h_i + h_{i-1}$
6: **for** $v_i \in V$ **do**
7: $\quad TC_i^\lambda \leftarrow h_i(G^\lambda) + Lap(\frac{2\lambda + 1}{\epsilon})$
return TC^λ

5 LOCAL CLUSTERING COEFFICIENT

In the last section we solved the problem of publishing the triangle counting with differential privacy. It is worth nothing that triangle counting is an important part to calculate the clustering coefficient and the transitivity ratio of a network, which is widely used in character recognition, spam detection, and community discovery. Thereby in this part we expand the privacy preserving mechanism of triangle counting to one of its applications: the local clustering coefficient. We first propose Alg. 4 using the local triangle counts to publish a safely preserved expression of the cohesion patterns in social networks.

Alg. 4 takes graph G and privacy budget ϵ as input, then draws the distribution of local clustering coefficient and the LC^λ-Histogram. This algorithm first collects the information of degree and triangle counts from each node in G, and then computes local clustering coefficient as below.

Definition 5.1. Given a graph $G = (V, E)$, we use N_i to represent the real triangles node v_i linked, and N_i' to represent most of the triangles it probably linked, and we have

$$Local\ Clustering\ Coefficient_i = N_i / N_i'$$

and for N_i' which is related to the degree of node v_i, so we have

$$N_i' = Deg(i) * (Deg(i) - 1)/2$$

where $Deg(i)$ represents the degree of node v_i.

Algorithm 4 LC^λ-Histogram: local clustering coefficient distribution

Input: Graph $G = (V, E)$ and privacy budget ϵ
Output: A noisy local clustering coefficient distribution TC^λ

1: $G^\lambda \leftarrow T_\lambda(G)$ by Alg. 1
2: **for** $v_i \in V$ **do**
3: $\quad Deg^\lambda(i) \leftarrow$ *Query: What is the degree of v_i?*
4: **for** $v_i \in V$ **do**
5: $\quad Tri^\lambda(i) \leftarrow$ *Query: How many triangles v_i links to?*
6: **for** $v \in V$ **do**
7: $\quad h_i(G^\lambda) \leftarrow \frac{2 * Tri^\lambda(i)}{(Deg^\lambda(i) * (Deg^\lambda(i) - 1)}$
8: **for** $v_i \in V$ **do**
9: $\quad LC_i^\lambda \leftarrow h_i(G^\lambda) + Lap(\frac{2\lambda + 1}{\epsilon})$
return LC^λ

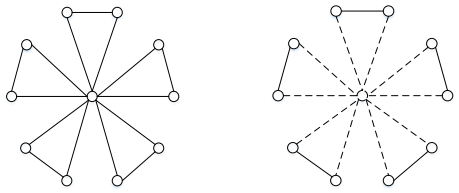
After computing the local clustering coefficient of each node, we get the distribution of the cohesion patterns of graph G. Such information needs to be perturbed before published. Therefore, we add Laplace noise to the local clustering coefficient distribution (shown in Alg. 4) with global sensitivity of $2\lambda + 1$, and its upper-boundary can be given as below.

Lemma 5.2. *Given any two graphs G and G′ differing by only one node, we have*

$$\| LC\text{-}hist(T_\lambda(G)) - LC\text{-}hist(T_\lambda(G')) \|_1 \leq 2\lambda + 1$$

Proof. From Definition 5.1 we can easily get that the local clustering coefficient has a range from 0 to 1. In the worst case all triangles in graph G' are connected to node v', that is to say, graph G has no triangle at all. To make the difference as large as possible, we can set all the nodes in graph G' except for v' that holds the local clustering coefficient of 1 (shown in Fig. 3). So that the degree of v' is $2m$ and its local clustering coefficient is $\frac{2m}{2m*(2m-1)}$. Now if we delete v' from G', the local clustering coefficient of all nodes except v' will change from 1 to 0. By adding up the difference of node v', the global sensitivity will be $2m + \frac{1}{2m-1}$. As we have defined $m \leq \lambda$ and clearly $m \geq 1$, finally we get the upper-boundary of global sensitivity as $2\lambda + 1$. □

In [23] the authors proposed a concept of *Outlink Privacy* to protect the local clustering coefficient, which can achieve low global sensitivity but lose the rigorous definition of differential privacy.



(a) the original graph (b) delete the center node

Figure 3: Deleting a node from a special graph may cause all triangles disappear

Different from it our goal is to protect the local clustering coefficient satisfying strict node-differential privacy. Using our proposed method in Section 4.1, we prove that we have the global sensitivity boundary of $2\lambda + 1$. In Definition 5.1, we can easily get that local clustering coefficients are all in the range of $[0, 1]$, and now we have the sensitivity of $2\lambda + 1$, which will cause noise in the same magnitude as λ. The noise is too generous to the local clustering coefficient distribution that it will make the noisy histogram useless. To solve this problem, we try to find a structure Ω, in which all coefficients are grouped into separate bins to publish the relationship between the coefficients and its corresponding node number. More specially, $\Omega = \{\tilde{g_1}, \tilde{g_2}, \tilde{g_3}, ..., \tilde{g_k}\}$ is a set of disjoint bins respecting the grouping strategy, where each $\tilde{g_i} = \{coe | c_1 \leq coe < c_2\}$ contains all coefficients in the interval of c_1 to c_2. For the first bin $\tilde{g_1}$, we have $c_1 = 0$, and for the last bin $\tilde{g_k}$, we have $\tilde{g_k} = \{coe | c \leq coe \leq 1\}$. To observe the coefficient distribution more intuitively, the interval $[0, 1]$ will be evenly divided into k groups, and each group corresponds to one bin in structure Ω.

Based on the coefficient grouping method, we can publish the distribution between the coefficient bins and the corresponding node number (LC^k-Histogram) to describe the statistical features between them, and we prove that this histogram has the global sensitivity $\Delta_{hist\text{-}coe}$ smaller than $4\lambda + 1$. Moreover, we can also publish the cumulative distribution of LC^k-Histogram, which we call it the CLC^k-Histogram. For this cumulative histogram, we prove that the global sensitivity $\Delta_{hist\text{-}ccoe}$ is no larger $2\lambda(k-1)+k$.

Lemma 5.3. *Given any two graphs G and G' differing only by one node and given the group number k, we have*

$$\| LC^k\text{-}hist(T_\lambda(G)) - LC^k\text{-}hist(T_\lambda(G')) \|_1 \leq 4\lambda + 1$$

where T_λ is the function in Alg. 1.

Proof. Following the mathematics symbol used in Lemma 4.2, and similarly node v' links to m triangles. When we delete v' from G', this will cause m triangles disappear. In the worst situation, these m triangles connect to $2m$ different nodes except v', and deleting v' will result in $2m$ nodes' local clustering coefficients changed. Projecting to the LC^k-Histogram, one node's variation can cause a difference of 2, therefore, $2m$ nodes will bring $4m$ difference in the histogram. We take the node v' we have deleted into consider, the total difference will be $4m + 1$. As we have set that $m \leq \lambda$, we have the global sensitivity $\Delta_{hist\text{-}coe}$ no larger than $4\lambda + 1$. □

Algorithm 5 LC^k-Histogram: local clustering coefficient distribution with k groups

Input: Graph $G = (V, E)$, privacy budget ϵ and group number k
Output: A noisy local clustering coefficient distribution LC^k

$G^\lambda \leftarrow T_\lambda(G)$ by Alg. 1
for $v \in V$ **do**
  $LC_i(G^\lambda) \leftarrow \frac{2 * Tri^\lambda(i)}{(Deg^\lambda(i) * (Deg^\lambda(i) - 1)}$
for $LC_i \in LC_i(G^\lambda)$ **do**
  if $LC_i = 1$ **then**
    $LC_i \in \tilde{g_k}$
  for j from 0 to $k - 1$ **do**
    if $LC_i \geq \frac{j}{k}$ and $LC_i < \frac{j+1}{k}$ **then**
      $LC_i \in \tilde{g_j}$
for $v \in V$ **do**
  $h_i(G^\lambda) \leftarrow$*Query: How many nodes that have the local clustering coefficient in the bin $\tilde{g_i}$?*
for $v_i \in V$ **do**
  $LC_i^k \leftarrow h_i(G^\lambda) + Lap(\frac{4\lambda + 1}{\epsilon})$
return LC^k

Lemma 5.4. *Given any two graphs G and G' differing only by one node and given the group number k, we have*

$$\| CLC^k\text{-}hist(T_\lambda(G)) - CLC^k\text{-}hist(T_\lambda(G')) \|_1 \leq 2\lambda(k-1) + k$$

where T_λ is the function in Alg. 1.

Proof. Similar to the proof of Lemma 4.3, we first consider the difference caused by v'. Assume that v' holds the coefficient of c_1 and $c_1 \in \tilde{g_1}$, then the difference caused by v' in this cumulative distribution will be k. As we know, deleting v' will influence at most $2m$ different nodes, and in the worst situation the local clustering coefficient of these $2m$ nodes may change from 1 to 0 (shown in Fig. 3). That is to say, in the CLC^k-Histogram the difference caused by the $2m$ nodes will be $2m * (k-1)$. Adding all the difference together, we get the global sensitivity for the CLC^k-Histogram is no larger than $2m * (k-1) + k$. Due to the premise that $m \leq \lambda$ and $k \geq 2$, we have the global sensitivity $\Delta_{hist\text{-}coe} \leq 2\lambda(k-1) + k$. □

Based on the theory of LC^k-Histogram, Alg. 5 adds the parameter k to the input and draws a noisy coefficient distribution as output. We first use Alg. 1 to get a preprocessed graph G^λ, whose triangle count can be limited to λ. Then we calculate the local clustering coefficient for each node and divide them into k different groups. After that we draw the distribution of these groups and the corresponding node numbers. Finally we add the Laplace noise to the distribution and get a noisy k-group coefficient distribution.

For cumulative k-group coefficient distribution we have proved that the global sensitivity upper-boundary can be $2\lambda(k-1) + k$, which changes influenced by both λ and k. When we set $k = 3$, the global sensitivity $\Delta_{hist\text{-}ccoe}$ is nearly equal to $\Delta_{hist\text{-}coe}$. However, when k is set larger, the noise we need to add to CLC^k-Histogram will be much larger than that to LC^k-Histogram. The experimental contrast between them will be discussed in the next section.

6 DATASETS AND EXPERIMENTS

6.1 Experimental Setup

Datasets. We used three real-world datasets available at [13]: Wiki-Vote (the network from Wikipedia of who-votes-on-whom), Cit-HepTh (Arxiv High Energy Physics paper citation network), and Twitter. Detailed statistics of each dataset is shown in Table 1. Here, Tri_{sum} is the sum of triangles in the dataset; Tri_{max} is the max triangle count for one node; Tri_{avg} is the average triangle count for one node. All directed graphs are pre-processed into undirected ones for easy implementation.

Table 1: Information of three datasets

Dataset	$\|V\|$	$\|E\|$	Tri_{sum}	Tri_{max}	Tri_{avg}
Wiki-Vote	7115	103689	608389	30940	86
Cit-HepTh	27770	352807	1478735	33527	53
Twitter	81306	1768149	13082506	96815	1698

Table 2: Triangles reserved after DL, DS, and DR

Dataset	Original	$\lambda = 512$		
		DL	DS	DR
Wiki-Vote	608389	**147649**	55422	80455
Cit-HepTh	1478735	**776619**	572934	642105
Twitter	13082506	**3284829**	1693775	2210566

Environment. We run all experiments on a PC with Intel Core i5-4590 @3.30GHz and 8GB RAM, a 64-bit Windows operating system. All codes were implemented in Java.

Parameters. We use L_1-distance and KS-distance to evaluate the utility of our proposed methods. For each approach we use the mean error of 100 runs. All privacy budget were set as $\epsilon \in [0.5, 1.5]$. We follow [2, 3, 20] and choose the threshold λ from a sequence of $\{1, 2, 4, 8, ..., 2^{\lfloor 2log_2(|V|) \rfloor}\}$.

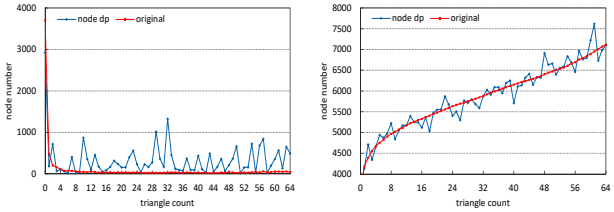
Histograms[1]**.** We proposed two kinds of histograms to publish triangle counts: Tr^{λ}-Histogram, which is a noisy triangle count distribution; TC^{λ}-Histogram, which is a noisy cumulative triangle count distribution. For local clustering coefficients we first divided them into k groups, then we published the LC^k-Histogram to show the noisy coefficient distribution and the CLC^k-Histogram to show the noisy cumulative coefficient distribution. Each node from all of them holds the triangle count upper-boundary of λ.

Performance for DL and DS with different choices of λ. To reserve as many triangles as possible when we try to get a λ-bounded graph $T_\lambda(G)$, we compared DL, DS, and DR that we proposed in Section 4.1, and the results of the three rules are shown in Table 2. Obviously, deleting edges randomly (says DR) can always achieve a medium effect between DL and DS, thus we focus more on the comparison of DL and DS. We found that the effect of DL and DS depend on the selection of λ. When λ is set very close to Tri_{max} (the max triangle count of one node), the number of triangles that DS reserved from the original graph may be a little more than that of DL. However, if we set λ much smaller than Tri_{max} (eg. $Tri_{max}/2$), DL can achieve much better effect than DS (shown in Table 2 column 3-4). Therefore, in this paper we choose DL as the rule to implement the preprocessing process for a λ-bounded graph $T_\lambda(G)$.

[1]Some of the negative noises in our experiment will make the value of bins in histograms lower than 0. Considering that in real world "a negative node number" is meaningless, we set these kind of bins as the value of 0. However, when we use the two utility metrics (Section 6.5) to evaluate the experiment results, we use the negative results other than setting them as 0.

6.2 Analysis of Tr^{λ}-Histogram and TC^{λ}-Histogram

For each triangle count distribution we compare the original real distribution and the noisy distribution. Using data from Wiki-Vote as an example, we find that the fluctuation of each dot caused by noise in Fig. 4(b) is much smaller than that in Fig. 4(a), which means that cumulative distribution achieves much better effect than triangle count distribution. This can be explained from their different sensitivities which are proved in Section 4. We prove that the sensitivity boundary of Tr^{λ}-Histogram is $4\lambda + 1$ and TC^{λ}-Histogram is $2\lambda+1$, that is to say, the noise added to the former one is almost two times larger to the latter one. Therefore, the cumulative distribution holds more similar shapes of original distribution than the triangle count distribution.



(a) triangle count distribution (b) cumulative distribution

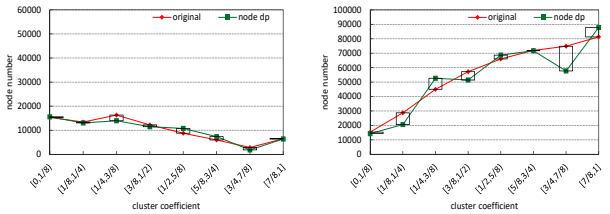
Figure 4: Comparison of triangle count distribution and cumulative distribution

6.3 Analysis of LC^k-Histogram and CLC^k-Histogram

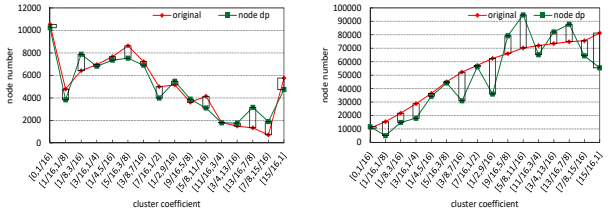
In Fig. 5 we set $\lambda = 512$ and compare the LC^k-Histogram and CLC^k-Histogram of Twitter with different parameter k. For each distribution we also compare the original real distribution (red line) and the noisy one (green line). Different from triangle count, when we keep k unchanged, the noise in cumulative coefficient distribution histogram is much lager than that in coefficient distribution histogram. That is to say, for local clustering coefficient the original distribution can achieve better effect than the cumulative one. In addition, we also find that different settings of parameter k will bring huge difference of noise in CLC^k-Histogram (shown in Fig. 5(a) and Fig. 5(c)). As we have proved in Section 5 that the coefficient distribution histogram holds the global sensitivity upper boundary of $4\lambda + 1$ while its cumulative distribution histogram of $2\lambda(k-1)+k$. As long as $k > 3$, the noise added to CLC^k-Histogram will be much more than that to LC^k-Histogram. Moreover, as k increases, the noise will increase linearly, which will lead to larger deviations. Thus in subsequent content we will focus on the analysis of experimental results on LC^k-Histogram.

6.4 Analysis of λ Selections

Different λ in a single dataset in TC^{λ}-Histogram. A larger threshold λ preserves more triangles from the original graph, however, it may bring larger noise to the published distribution. Fig. 6

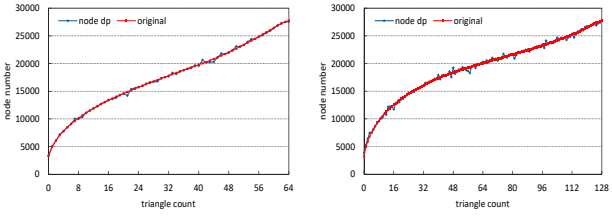


(a) coefficient distribution (k=8) (b) cumulative distribution (k=8)

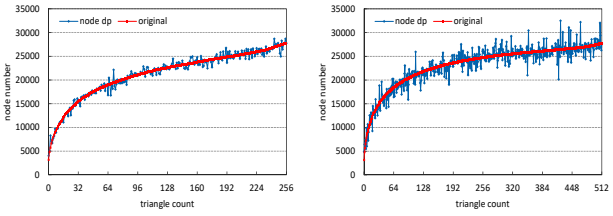


(c) coefficient distribution (k=16) (d) cumulative distribution (k=16)

Figure 5: Comparison of coefficient distribution and cumulative distribution



(a) λ=64 (Cit-HepTh) (b) λ=128 (Cit-HepTh)



(c) λ=256 (Cit-HepTh) (d) λ=512 (Cit-HepTh)

Figure 6: Comparison of different λ in dataset Cit-HepTh

shows different selections of λ when we publish TC^{λ}-Histogram of Cit-HepTh. When we choose $\lambda = 64$ or 128, the noisy distributions are almost the same as the real ones. But as the threshold is set up to 256 or even 512, the amplitude of noise becomes greater. As we can see in Fig. 6(d), the noise has beyond our limit. So for Cit-HepTh, the best choice of λ can be in the range of 128 to 256, which can guarantee the privacy properly while save the statistical characteristics of the real distributions.

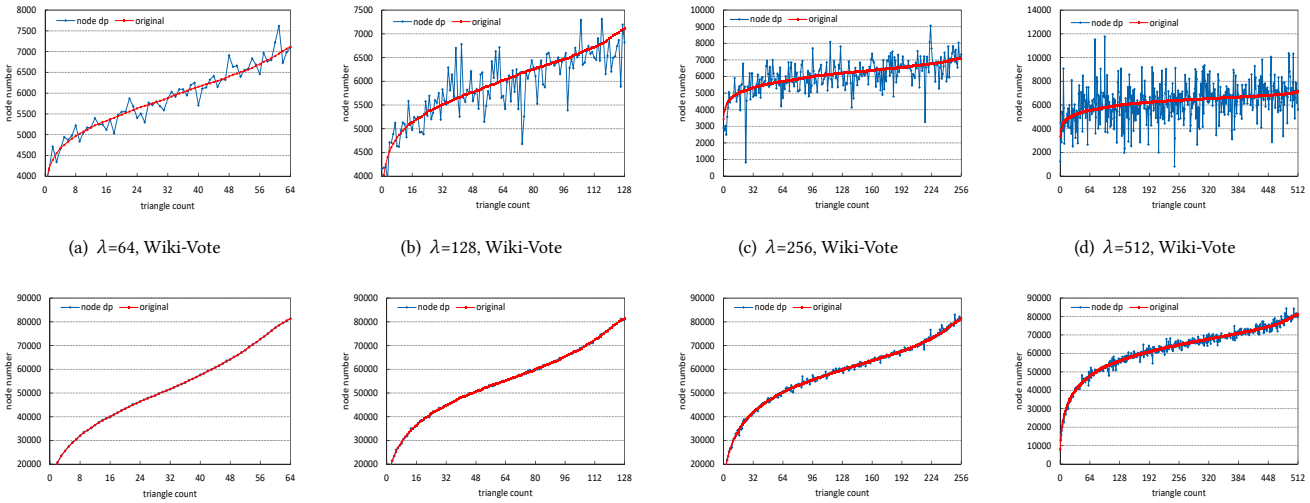
Selection of λ for small & large datasets in TC^{λ}-Histogram. In Fig. 7 we compare the cumulative distribution of Wiki-Vote (7115 nodes) and Twitter (81306 nodes) with different λ from our proposed function in Alg. 1. We can find that a larger λ is much more suitable for a larger dataset. When we choose $\lambda = 128$ or 256 for Twitter, the noisy distribution and the original real one are almost completely overlapped. While the effect of privacy protection is very good, we have lost more information than we could have saved. In comparison, when we choose $\lambda = 512$ for Twitter, the comprehensive evaluation of our noisy distribution becomes much better. In addition, if we choose a larger threshold for a relatively small dataset, it may make the noisy distribution become useless (shown in Fig. 7(b)). Since a smaller dataset is more possible to have less triangles (shown in Table 1), and for Wiki-Vote, the average triangle count is 86, which is much smaller than the threshold 256 or 512, so set $\lambda = 128$ or smaller will be more appropriate for Wiki-Vote. In general, for datasets with different orders of magnitude, set different and suitable thresholds is a more effective way to improve the availability of our methods.

Selection of λ and k for LC^{k}-Histogram. Experimental results of different λ and k settings of LC^{k}-Histogram are presented in Fig. 8. The left half of this figure holds the parameter $k = 8$, while the right half holds $k = 16$. In each of the half we compare different λ settings of 128 and 256 among the three datasets. When we compare the figures vertically, we find that a lager dataset can more probably obtain a better effect in LC^{k}-Histogram. When we explore the function of parameter k, we compare the first and third columns (or the second and fourth columns) and conclude that a larger k can display the polyline moving trends more detailedly. In the coefficient distributions we also find that the noises added to histogram will grow as the threshold λ increases, and the principle is just similar to the triangle count distribution.

6.5 Analysis of Utility Function

We use L_1-distance to evaluate our triangle count distribution and KS-distance to evaluate TC^{λ}-Histogram (shown in Fig. 9) and LC^{k}-Histogram (shown in Fig. 10). The result of utility function for TC^{λ}-Histogram shows that for one dataset, most of the time we get larger L_1-distance and KS-distance with a smaller ϵ (this is not absolute because the noise is random). From Fig. 9(b) and Fig. 10(b) we can see that KS-distance of the two distributions are all in a small magnitude, which shows that most of our noisy histograms hold extremely similar shapes from the original distribution, and our proposed method has achieved quite good effect on protecting privacy among these datasets.

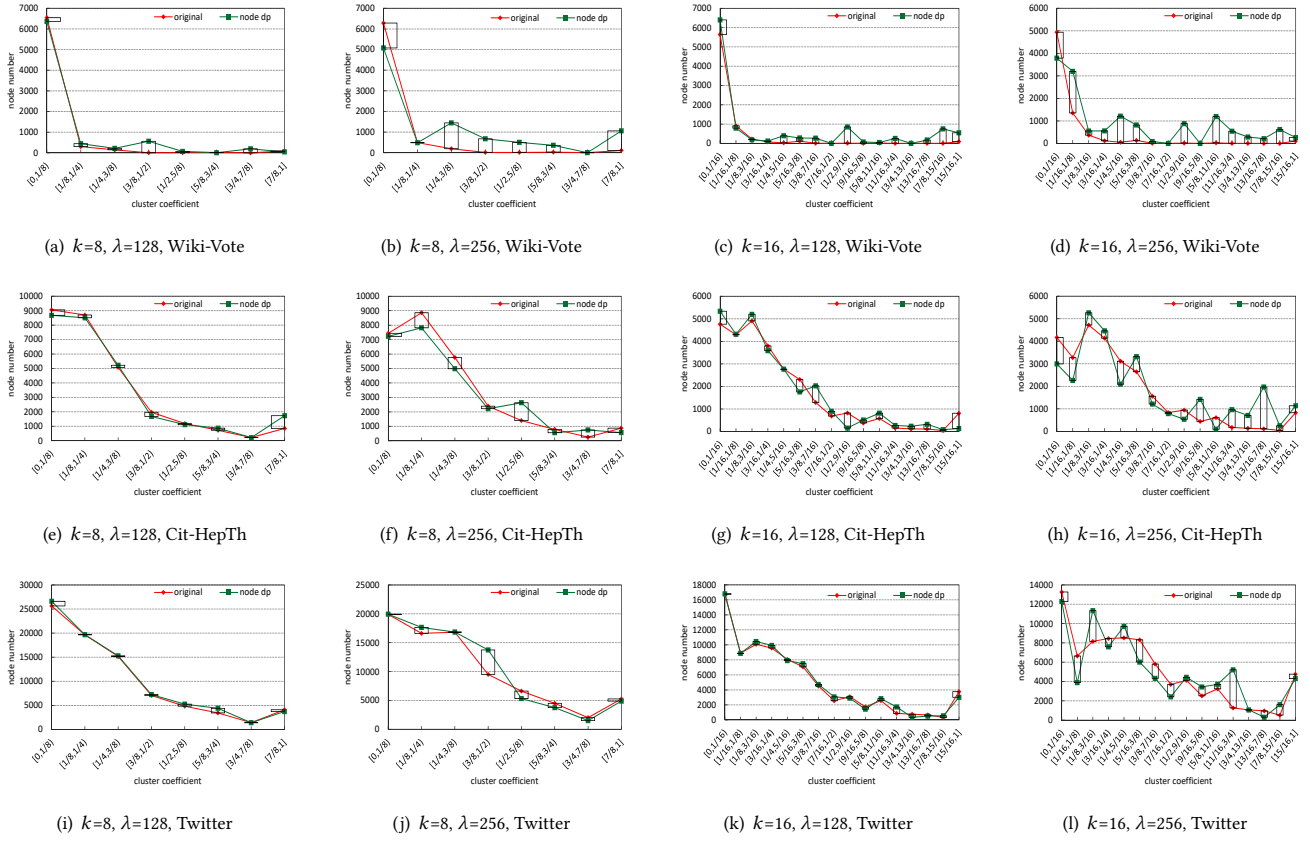
We also test the utility function of TC^{λ}-Histogram on one dataset for different λ (shown in Fig. 11), and the result shows that larger λ and smaller ϵ are more probably to bring larger errors both in the L_1-distance and KS-distance, corresponding to the principle of differential privacy. In Fig. 11 we can see that for Cit-HepTh, choosing $\lambda = 512$ will significantly increase L_1-distance and KS-distance, which means that the threshold is too large. This also verified the conclusion we obtained from Fig. 6. Besides the influence of λ, different ϵ can also impact the errors, and this confirms the conclusion in Fig. 9. However, due to the randomness of the noise added to each bin of histogram, and the definition of KS-distance (compute the "max" distance), this error may become small (shown in Fig. 11(b) when λ=512 and ϵ=1) when ϵ is set smaller.



(a) λ=64, Wiki-Vote (b) λ=128, Wiki-Vote (c) λ=256, Wiki-Vote (d) λ=512, Wiki-Vote

(e) λ=64, Twitter (f) λ=128, Twitter (g) λ=256, Twitter (h) λ=512, Twitter

Figure 7: Different selections of λ for small & large datasets

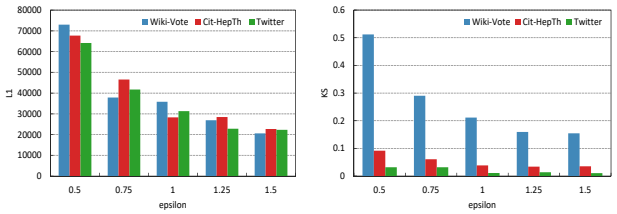


(a) k=8, λ=128, Wiki-Vote (b) k=8, λ=256, Wiki-Vote (c) k=16, λ=128, Wiki-Vote (d) k=16, λ=256, Wiki-Vote

(e) k=8, λ=128, Cit-HepTh (f) k=8, λ=256, Cit-HepTh (g) k=16, λ=128, Cit-HepTh (h) k=16, λ=256, Cit-HepTh

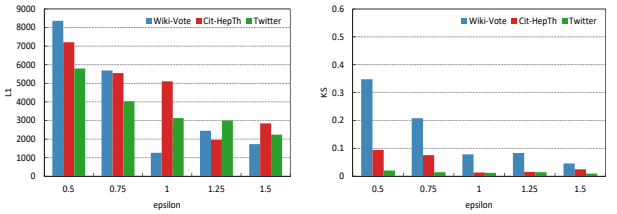
(i) k=8, λ=128, Twitter (j) k=8, λ=256, Twitter (k) k=16, λ=128, Twitter (l) k=16, λ=256, Twitter

Figure 8: Different selections of λ and k for LC^k-Histogram



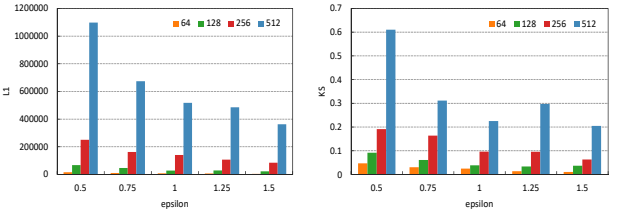
(a) L_1 (λ=128) (b) KS (λ=128)

Figure 9: Utility metrics for each dataset in TC^{λ}-Histogram



(a) L_1 (k=8, λ=128) (b) KS (k=8, λ=128)

Figure 10: Utility metrics for each dataset in LC^{k}-Histogram



(a) L_1 of Cit-HepTh (b) KS of Cit-HepTh

Figure 11: Utility metrics for Cit-HepTh with different λ

7 CONCLUSIONS

In this paper, we introduced a T_{λ} algorithm to anonymize triangle counts on large graphs using the node-differential privacy. We proposed two approaches: Tr^{λ}-Histogram and TC^{λ}-Histogram to publish triangle counts. To satisfy the strict node-differential privacy, we proved that the upper-boundaries of global sensitivity for them are $4\lambda + 1$ and $2\lambda + 1$ respectively. In addition, we analyzed and proved that our cumulative distribution TC^{λ}-Histogram can achieve much better effect than the triangle count distribution Tr^{λ}-Histogram. Furthermore, we extend the research of triangle count to the local clustering coefficient and published it by dividing the coefficients into k different groups. For coefficients we also proposed two approaches: LC^{k}-Histogram and CLC^{k}-Histogram and provided the sensitivities for them and proved them, which are $4\lambda + 1$ for the former and $2(k-1)\lambda + k$ for the later. Finally, we used L_1-distance and KS-distance to evaluate our proposed approaches. Experimental results show that our proposed data publishing methods through T_{λ} have the advantages of low privacy sensitivity and high data utility, future researches are required to be better investigated on how to choose the proper value of λ for each graph and discussed the strategy for k selections for different datasets.

8 ACKNOWLEDGMENTS

This work is supported in part by the National Key Research and Development Program of China under grant No. 2018YFB1004002 and the National Science Foundation of China under grant 61472148. Zhifeng Bao was supported in part by ARC (DP170102726, DP180102-050), NSFC (61728204, 91646204).

REFERENCES

[1] Z. Chang, L. Zou, and F. Li. Privacy preserving subgraph matching on large graphs in cloud. In *Proceedings of SIGMOD*, pages 199–213, 2016.
[2] S. Chen and S. Zhou. Recursive mechanism: Towards node differential privacy and unrestricted joins. In *Proceedings of SIGMOD*, pages 653–664, 2013.
[3] W. Y. Day, N. Li, and L. Min. Publishing graph degree distribution with node differential privacy. In *Proceedings of SIGMOD*, pages 123–138, 2016.
[4] H. G. Do and W. K. Ng. Privacy-preserving triangle counting in distributed graphs. In *Proceedings of IEEE AINA*, pages 917–924, 2016.
[5] C. Dwork. Differential privacy. In *Proceedings of ICALP*, pages 1–12, 2006.
[6] C. Dwork, F. Mcsherry, and K. Nissim. Calibrating noise to sensitivity in private data analysis. In *Proceedings of TCC*, pages 265–284, 2006.
[7] A. Farasat, G. A. Gross, R. Nagi, and A. G. Nikolaev. Social network analysis with data fusion. *IEEE Trans. Comput. Social Systems*, 3(2):88–99, 2016.
[8] X. Hu, Y. Tao, and C.-W. Chung. Massive graph triangulation. In *Proceedings of SIGMOD*, pages 325–336, 2013.
[9] A. Itai and M. Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7 (4):413–423, 1978.
[10] Z. Jorgensen, T. Yu, and G. Cormode. Publishing attributed social graphs with formal privacy guarantees. In *Proceedings of SIGMOD*, pages 107–122, 2016.
[11] V. Karwa and A. B. Slavković. Differentially private graphical degree sequences and synthetic graphs. In *Proceedings of PSD*, pages 273–285, 2012.
[12] S. P. Kasiviswanathan, K. Nissim, and S. Raskhodnikova. Analyzing graphs with node differential privacy. In *Proceedings of TCC*, pages 457–476, 2013.
[13] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.
[14] C. Li, M. Hay, V. Rastogi, G. Miklau, and A. Mcgregor. Optimizing linear counting queries under differential privacy. In *Proceedings of SIGMOD*, pages 123–134, 2010.
[15] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of ICDE*, pages 106–115, 2007.
[16] P. Li, H. Dau, G. Puleo, and O. Milenkovic. Motif clustering and overlapping clustering for social network analysis. In *Proceedings of INFOCOM*, pages 1–9, 2017.
[17] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *Proceedings of SIGMOD*, pages 93–106, 2008.
[18] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: privacy beyond k-anonymity. In *Proceedings of ICDE*, page 24, 2006.
[19] F. Mcsherry and K. Talwar. Mechanism design via differential privacy. In *Proceedings of FOCS*, pages 94–103, 2007.
[20] S. Raskhodnikova and A. D. Smith. Efficient lipschitz extensions for high-dimensional graph statistics and node private degree distributions. *CoRR*, abs/1504.07912, 2015.
[21] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.
[22] C. Tai, P. S. Yu, D. Yang, and M. Chen. Privacy-preserving social network publication against friendship attacks. In *Proceedings of SIGKDD*, pages 1262–1270, 2011.
[23] C. Task and C. Clifton. What should we protect? defining differential privacy for social network analysis. *In State of the Art Applications of Social Network Analysis. Springer LNSN*, pages 139–161, 2014.
[24] D. J. Watts and S. H. Strogatz. Collective relaxation dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.
[25] A. W. Wolfe. Social network analysis: Methods and applications. *American Ethnologist*, 24(1):219–220, 2010.
[26] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. *IEEE Trans. Knowl. Data Eng.*, 23(8):1200–1214, 2011.
[27] J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu. Differentially private histogram publication. In *Proceedings of ICDE*, pages 32–43, 2012.
[28] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. Private release of graph statistics using ladder functions. In *Proceedings of SIGMOD*, pages 731–745, 2015.
[29] X. Zhang. Towards accurate histogram publication under differential privacy. In *Proceedings of SDM*, pages 587–595.
[30] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *Proceedings of ICDE*, pages 506–515, 2008.