Edge Anonymity in Social Network Graphs

Lijie Zhang and Weining Zhang
Department of Computer Science, University of Texas at San Antonio
{lijez, wzhang}@cs.utsa.edu

***Abstract*—Edges in social network graphs can model sensitive relationships. In this paper, we consider the problem of edges anonymity in graphs. We propose a probabilistic notion of edge anonymity, called graph confidence, which is general enough to capture the privacy breach made by an adversary who is able to pinpoint target persons in a graph partition based on any given set of topological features of vertexes. We then focus on a special type of edge anonymity problem which uses vertex degree to partition a graph. We analyze edge disclosure in real-world social networks and show that even if graphs are anonymized to prevent vertex disclosure, they still do not guarantee edge anonymity. We present three heuristic algorithms that protect edge anonymity using edge swap or edge deletion. Our experimental results, based on three real-world social networks and several utility measures, show that these algorithms can effectively protect edge anonymity and obtain anonymous graphs of acceptable utility.**

I. Introduction

Social networks emerge as an important platform for people to establish, discover, and maintain their relationships with others. These systems also provide enormous potential for e-business and present unique opportunities for social behavior research. Conceptually, social networks are graphs in which vertexes represent individuals and edges represent relationships among individuals, such as friendship, trust, and social contact. These graphs are extremely useful for studying patterns of social influence [1], models of viral marketing [9], collaborative filtering in recommendation systems [2], to name a few. However, these applications of social networks also raise serious concerns of privacy.

It has been pointed out [3], [7] that privacy can be breached in published social networks even if they do not contain personal identities. One type of attack to privacy is vertex re-identification [7], [11], [18], [19] whereby an adversary can identify the vertex of a target person by analyzing topological features of the vertex based on his background knowledge about the person. For example, based on his knowledge about a target person Alice, the adversary may know the degrees of vertexes of Alice and her friends in a social network. Using this knowledge, the adversary can analyze a published graph of the social network and identify all vertexes whose surrounding graphs (SGs) have these specific degrees. These vertexes form an equivalence class (EC). Alice can be re-identified if her equivalence class contains a single vertex. A fundamental privacy notion with respect to vertex re-identification is vertex k-anonymity, which requires each equivalence class to contain at least k vertexes.

Another type of attack is edge re-identification [12], [16], [17], [19] whereby the adversary can identify the edge incident to vertexes of two target persons by analyzing topological features of the vertexes using his knowledge about the pair of persons. Edge re-identification is a breach to privacy because edges may represent sensitive information. Notice that edge can be re-identified even when vertexes cannot. For example, the adversary may know the degrees of two target persons and use this knowledge to identify ECs of the two persons. If every vertex in one EC has an edge with every vertex in the other EC, the adversary can infer with probability 1.0 that an edge exists between the two target persons, even if the adversary may not be able to identify the two persons within their respective ECs. Unlike vertex anonymity, there is currently no well-established privacy notion of edge anonymity.

A. Our Contributions

In this paper, we focus on protecting edge anonymity in published social networks. First, we propose a probabilistic privacy notion of edge anonymity, called graph confidence, which is defined with respect to a partition the graph using a vertex description type (VDT), topological features of vertexes. This privacy notion captures the privacy breach made by an adversary who is able to pinpoint vertex ECs of target persons in a graph partition based on any given set of topological features. Although it is an open problem to determine what is the most appropriate assumption of the ability of the adversary, our notion of edge anonymity is general enough to accommodate any given VDT. We then focus on a special type of edge anonymity problem, which uses vertex degree as the VDT, and make the following contributions.

1) We analyze edge disclosure in two real-world social networks and show that edge disclosure can occur in these graphs, especially the denser one, both before and after applying algorithms of vertex anonymity.
2) We present three algorithms to obtain τ-confidence graphs, graphs whose confidence is no less than a threshold τ. These algorithms use heuristics to perform either edge swap or edge deletion, so that not only to achieve edge anonymity but also to preserve utility.
3) We study empirically the performance and utility of these algorithms. Our preliminary results, based on three real-world social networks and several utility measures, show that these algorithms can effectively protect edge anonymity and can obtain anonymous graphs of acceptable utility.

B. Related Work

Several methods have been proposed to prevent vertex re-identification through vertex k-anonymity, which places at least k vertexes in each equivalence class. These methods differ in the types of the structural features that an adversary might use to partition vertexes in anonymous graphs.

Liu and Terzi [11] studied the vertex re-identification attack assuming that the adversary knows only the degree of the vertex of a target person. Their method obtains a vertex k-anonymous graph by adding or deleting edges of the original graph, so that, there are at least k vertexes of each degree.

Zhou and Pei [18] studied a *neighborhood attack,* in which the adversary re-identifies a target person by partitioning the graph according to isomorphism of vertex neighborhoods, a type of subgraph surrounding a vertex. Their method obtains a vertex k-anonymous graph by adding edges, so that, at least k vertexes will have isomorphic neighborhoods.

Hay et al. [7] model the background knowledge of an adversary as the answer to a knowledge query and proposed a method to prevent a vertex re-identification attack based on any type of knowledge queries. Their method obtains a vertex k-anonymous supergraph by aggregating vertexes into supernodes and edges into superedges, so that, each supernode represents at least k vertexes and each superedge represents all edges between vertexes in two supernodes. Since supernodes and superedges do not reveal internal topology, users of a supergraph have to generate conventional graphs by random sampling.

A number of methods were also proposed for edge anonymity.

Singh and Zhan [14] briefly introduced edge inference breach and proposed to measure graph anonymity in terms of vertex degree and vertex clustering coefficient. But they did not measure the privacy between individuals.

Zheleva et al. [17] considered a link re-identification attacks in which the adversary infers sensitive relationships from non-sensitive ones in graphs that contains multiple types of edges. In addition to removing all sensitive edges, their method also removes some non-sensitive edges and aggregates vertexes/edges into clustered nodes/edges.

Ying and Wu [16] considers edge re-identification attacks in which the adversary does not have any background knowledge. Their methods obtain an anonymous graph by performing random edge swap, edge addition, and edge deletion. However, since their methods typically introduce very small random noise, the anonymous graphs obtained by these methods may not provide sufficient protection to edge anonymity.

We refer interested readers to the recent survey by Liu et al. [10], which gives a more detailed account on research of privacy-preserving data analysis on graphs and networks.

C. Road maps

The rest of the paper is organized as follows. In Section II, we define the notion of graph confidence. In Section III, we analyze edge disclosure in graphs of two real-world social networks. In Section IV, we present three heuristic algorithms for edge anonymity. In Section V, we present experimental results on utility and performance of our algorithms. Section VI draws conclusions.

II. Edge Anonymity

In the following, we define a privacy notion of edge anonymity, which is based on a partition of graph.

A. Vertex Description Type and Graph Partition

Let $G = (V, E)$ be a simple undirected graph, where $V = \{v_1, \ldots, v_n\}$ is a set of vertexes and $E = \{(v_i, v_j) | v_i, v_j \in V\}$ is a set of edges. Each vertex represents an unidentified person. For convenience, we will not distinguish vertex from person.

Definition 1: (**radius-r subgraph**) *For an integer $r \geq 0$, the* radius r subgraph *(or r-SG for short) of a vertex $v \in V$ is $sg_r(v) = (V', E')$, where $V' \subseteq V$ such that $\forall v' \in V'$ the shortest path between v and v' contains at most r edges, and $E' \subseteq E$ consists of edges among vertexes in V'.*
Many topological descriptions of graphs have been defined in graph theory [15], including degree sequence, adjacency matrix, shortest path, hub, etc. A topological description of the subgraph surrounding a vertex can give a non-trivial description of the vertex.

Definition 2: (**vertex description type**) *A* vertex description type *(VDT) D is a set of finite topological descriptions of r-SGs of a given radius $r \geq 0$, with an equivalence relation $\sim_D$ and a function that maps each vertex v to $D(v)$, the type-D description of vertex v.*
For example, the set of degrees of vertexes is a VDT (here $r = 0$) whose equivalence relation is the integer equality; the set of degree sequences of 1-SGs with the equality of sequence of integers up to a permutation as the equivalence relation is another VDT; and a set of adjacency matrices of 3-SGs with the graph isomorphism is yet another VDT. Implicitly, VDTs have been used in previous studies of graph anonymization. Examples include vertex degree [7], [11], hubs [7], and neighborhood [18].

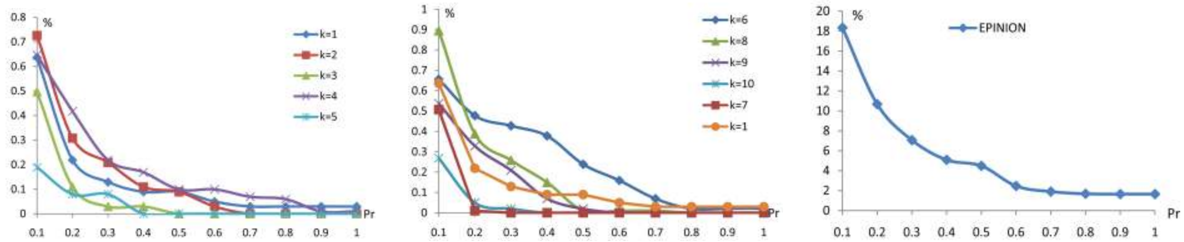
Definition 3: (**graph partition**) *Let D be a VDT*[1]. *The* type-D vertex partition *of a graph $G = (V, E)$ is $\mathcal{P}_D(G) = \{C | C \subseteq V, (\forall v_1, v_2 \in C)(D(v_1) \sim_D D(v_2))\}$, where each $C \in \mathcal{P}_D(G)$ is a vertex equivalence class (or VEC). The* type-D edge partition *of G is $\{E_{ij} | (\forall (v, v') \in E_{ij})(v \in C_i, v' \in C_j, C_i \in \mathcal{P}_D(G), C_j \in \mathcal{P}_D(G)\}$, where E_{ij} is an edge equivalence class (or EEC) with end VECs C_i and C_j.*
Intuitively, a VEC C contains all vertexes of an equivalent type-D description $D(C)$ and an EEC E_{ij} contains all edges between VECs C_i and C_j.

B. Graph Confidence

Suppose that an adversary is able to determine in a graph partitioned using a given VDT two (not necessarily distinct) VECs for target persons u and v, but cannot distinguish these target persons from other persons in their VECs. The adversary wants to determine the probability that there is an edge linking the two individuals.

Definition 4: (**linking probability**) *Let C_i and C_j be (not necessarily distinct) VECs in a type-D vertex partition of a*

[1] For the sake of presentation, we only consider vertex partitions based on a single VDT. However, our results apply to partitions based on multiple VDTs (such as $\mathcal{P}_{D_1,D_2}(G)$).



(a) COA $k = 1$ to 5

(b) COA $k = 6$ to 10

(c) EPINION

Fig. 1: Edge Disclosure in Social Networks

graph G. The probability that an edge in E_{ij} links a target person u hidden in C_i and a target persons v hidden in C_j is

$$p_{ij} = Pr[C_i, C_j] = \frac{\alpha_{ij}}{\beta_{ij}} \qquad (1)$$

where α_{ij} is the number of edges in E_{ij}, i.e., $\alpha_{ij} = |E_{ij}|$ and β_{ij} is the number of pairs of vertexes between C_i and C_j, that is,

$$\beta_{ij} = \begin{cases} \frac{|C_i| \times (|C_i|-1),}{2} & i = j; \\ |C_i| \times |C_j|, & i < j. \end{cases}$$

Intuitively, β_{ij} is the number of possible relationships between individuals in C_i and individuals in C_j. Since only α_{ij} out of β_{ij} relationships actually exists between C_i and C_j, the probability that any randomly chosen pair of individuals has a relationship is $\frac{\alpha_{ij}}{\beta_{ij}}$, the ratio of actual edges and possible edges between C_i and C_j, assuming that each pair of individuals is equally likely to have a relationship.

Definition 5: (τ-**confidence**) *Given a VDT D. The confidence of a graph G (that it protects edge anonymity) is defined as $conf_D(G) = 1 - \max P_{G,D}$, where $P_{G,D} = \{p_{ij} | C_i, C_j \in \mathcal{P}_D(G), i \leq j\}$ is the set of linking probabilities calculated based on the type-D partition of G. A graph G is τ-confident wrt D if $conf_D(G) \geq \tau$.*

C. Discussion

Definition 4 assumes that the adversary has complete knowledge to locate a single VEC for each target person. In practice, the adversary may only have partial knowledge and cannot locate a single VDT. For example, the adversary may know that target persons have certain degrees and the anonymous graph is obtained by deleting edges of the original graph, but does not know if the degrees of target persons in the anonymous graph are reduced or not. In this case, if vertex degree is used as the VDT for graph partition, the adversary has to assume that each target person may be hidden in any VEC whose degree is equal to or less than the true degree of the person. Definition 4 can be easily extended as follows to cover this case.

Definition 6: (**linking probability generalized**) *Let S_u and S_v be two sets of VECs in a type-D vertex partition of a graph. The probability that there is an edge linking a target person u hidden in S_u and a target person v hidden in S_v is*

$$Pr[S_u, S_v] = \frac{1}{|S_u| \times |S_v|} \sum_{C_i \in S_u, C_j \in S_v} Pr[C_i, C_j].$$

This definition assumes that target person u (or v) is equally likely to be hidden in any VEC in S_u (or S_v). With Definition 6, if an anonymous graph is τ-confident with respect to (wrt) single VEC under a VDT, it is also τ-confident wrt multiple VECs. This is obvious because if $\forall C_i, C_j \in \mathcal{P}_D(G)$, $Pr[C_i, C_j] \leq 1 - \tau$, then $\forall S_u, S_v \subseteq \mathcal{P}_D(G)$, $Pr[S_u, S_v] \leq 1 - \tau$. Thus, in the rest of this paper, we only consider confidence wrt single VEC.

Our definition of graph confidence is general in the sense that it allows for any given VDT. Of course, the more complex the VDT is, the more powerful the adversary is, and the more difficult the edge anonymity problem becomes. However, we believe that in practice, it is not necessary to use very complex VDT. The reason is that if the adversary is able to use a very complex VDT to attack, he perhaps already knows the sensitive information targeted by the very attack. Nonetheless, it is still an open problem to determine which VDT is the most appropriate for protecting edge anonymity. We leave it for future research. In the rest of this paper, we focus on a special type of edge anonymity problem that uses vertex degree as the VDT.

III. Edge Disclosure in social Networks

In this section, we analyze through experiments the risk of edge disclosure in some real-world social networks. We consider two datasets: EPINION[6] and COA [4]. The EPINION dataset is the "web of trust" social network extracted from Epinion.com. The dataset is a directed graph in which vertexes represent members and edges represent the trust relationship among members. For the purpose of our experiments, we converted the data into an undirected graph by simply ignore the direction of edges. The resulting graph contains 49,287 vertexes and 381,035 edges. The COA dataset is a social network used in [11]. The dataset is an undirected graph containing 7,955 vertexes and 10,055 edges.

To investigate edge disclosure in graphs produced by algorithms of vertex anonymity, we implemented a vertex k-anonymity algorithm described in [11]: the *priority* algorithm with the probing scheme using degree as a VDT and edge deletion as anonymization strategy. We applied this algorithm on COA graph to generate anonymous graphs. However, we were unable to obtain vertex anonymous graph of EPINION dataset using this algorithm due to the size and density of the graph. In fact, all existing vertex k-anonymity algorithms have some problems working on EPINION. For example, we also

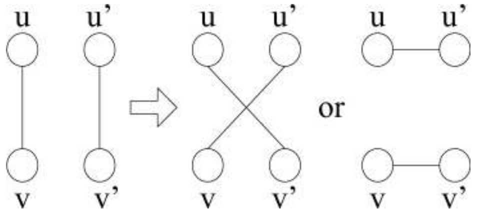


Fig. 2: Edge Swap

implemented the simulated annealing algorithm of [7], which searches for anonymous graph that optimizes a likelihood estimate. However, the computation of the likelihood of a single graph is $O(|E|)$. Due to the size of EPINION graph, the computation for split or moving one node can cause an out-of-memory exception on a computer with 2GB main memory and can take 23 minutes on a computer with 128GB main memory.

We partitioned each graph using degree as the VDT, measured the linking probabilities of each EEC, and counted percentage of edges of various linking probabilities. The results are shown in Figure 1, in which the X-axis is the linking probability and the Y-axis is the percentage of edges. In Figures 1a and 1b, curves correspond to vertex k-anonymous graphs of COA, where $k = 1$ corresponds to the original graph. A point (x, y) on a curve means that y percent of edges have a linking probability of at least x. Based on the results, we have the following observations.

1) Edge disclosure is more likely to occur in dense graphs. As shown in Figure 1c, in EPINION graph, 5% (or 17064) edges have a linking probability of 0.5 or higher, and 2% (or 6280) edges are completely disclosed. These numbers become 0.1% (9) and 0.03% (3), respectively, in COA graph.
2) Edge disclosure can still occur in vertex k-anonymous graphs. As shown in Figures 1a and 1b, even though the original COA graph has much less risk of edge disclosure than EPINION, the vertex k-anonymity algorithm still cannot protect edge anonymity. Interestingly, for $k = 4$ or 6, the anonymous graphs have higher risk of edge disclosure than the original graph.

Based on this analysis, we believe that algorithms specifically designed for edge anonymity are needed.

IV. Algorithms for Degree-based Edge Anonymity

Even for the special type of edge anonymity problem that uses degree of vertex as the VDT, finding the optimal τ-confident edge anonymization is intractable. Therefore, in this section, we present heuristic algorithms.

There are four graph anonymization strategies: 1) random edge deletion, 2) random edge addition, 3) random edge swap and 4) random edge addition/deletion. Edge swap is a special type of edge addition/deletion that deletes two edges and adds back two new edges connecting the four vertexes in one of the two specific ways illustrated in Figure 2. A basic operation of each strategy makes a minimum change to a graph. For example, deleting one edge is a basic operation of edge deletion, and swapping two edges is a basic operation of edge swap.

Algorithm 1 Degree-based Edge Swap

```
Input: graph $G = (V, E)$ and threshold $\tau$
Output: $\tau$-confident graph $G'$
Method:
    $G' = G$;
    partition $G'$ by vertex degree;
    while (confidence of $G'$ is less than $\tau$) do
        randomly select an edge $e_1$ from the leading EEC;
        find second edge $e_2$ according to Theorem 1;
        if $e_2$ exists, perform edge swap with $e_1$ and $e_2$;
        else return an empty graph;
    end while
    return $G'$;
```

For edge anonymity, these strategies have different impact on linking probability. For example, edge deletion can always reduce linking probability, but edge addition may increase linking probability. On the other hand, the effect of edge swap is often difficult to predict. These anonymization strategies also have different impact on different graph measurements [5]. For example, edge swap does not alter vertex degree, but may change centrality and shortest paths.

In the rest of this section, we present algorithms that perform either edge swap or edge deletion. Intuitively, it is also possible to obtain edge anonymity by adding edges. However, directly adding counterfeit edges to the original graph will not help, because adding more edges to EECs will increase linking probabilities. (Although the added edges do not represent real relationships, they cause those real relationships to be identified more easily.) One option will be to start from the graph that contains all vertexes but no edge, and then add original edges into the graph one at a time as long as the edge anonymity requirement is still satisfied. To preserve utility, we will want to add back to the graph as many original edges as possible. This method however can suffer from poor performance if the original graph contains a large number of vertexes and edges, and the majority of original edges will remain in the anonymous graph. Thus, for practical reasons, we do not consider algorithms that use edge addition.

A. *Degree-based Edge Swap*

Algorithm 1 takes as input a graph and a confidence threshold, and used edge swap to obtain a τ-confident anonymous graph if one can be found or an empty graph otherwise. The goal is to find a graph that not only satisfies the privacy requirement but also has a good utility. To achieve this goal, the algorithm uses a greedy strategy to improve graph confidence, namely, it focuses on reducing the size of the *leading EEC*, which corresponds to the maximum linking probability of the graph. Intuitively, reducing the size of the leading EEC may improve graph confidence more quickly than reducing the size of other EECs, therefore result in fewer edges being swapped and better utility of anonymous graphs.

Algorithm 2 Edge Deletion with Maximum Choice

```
Input: graph $G = (V, E)$ and threshold $\tau$
Output: $\tau$-confident graph $G'$
Method:
 1.  $G' = G$;
 2.  partition $G'$ by vertex degree;
 3.  while (confidence of $G'$ is less than $\tau$) do
 4.     for each edge $e$ in the leading EEC do
 5.        compute RMLP and IOLP of deleting $e$;
 6.     delete the edge of maximum RMLP and
           minimum IOLP;
 7.  end while
 8.  return $G'$;
```

Algorithm 3 Edge Deletion with Random Choice

```
Input: graph $G = (V, E)$ and threshold $\tau$
Output: $\tau$-confident graph $G'$
Method:
 1.  $G' = G$;
 2.  partition $G'$ by vertex degree;
 3.  while (confidence of $G'$ is less than $\tau$) do
 4.     randomly delete an edge in the leading EEC;
 5.  return $G'$;
```

In each iteration (steps 3-8), the algorithm attempts to swap the pair of edges that can lead to the biggest improvement of the graph confidence. If such a pair of edges does not exist, the algorithm will terminates with an empty graph. To choose the edges to swap, Algorithm 1 takes an edge from the leading EEC and find a second edge from an EEC that satisfies the conditions of the following theorem.

Theorem 1: Let a graph G be a graph partitioned using vertex degree and G' be the graph obtained by a valid swap of two edges $e_1 \in E_{ij}$ and $e_2 \in E_{st}$. where $i \leq j$, $s \leq t$. Then, for each EEC E_{xy} in G' that receives any new edge, the corresponding linking probability p'_{xy} is less than p_{ij} if and only if indexes i, j, s, t contain at least two distinct values and each of these values appears at most twice, and one of the following conditions holds.

1) *$i = j$, $s = t$, and $\alpha_{is} < \alpha_{ii} \cdot \frac{\beta_{is}}{\beta_{ii}} - 2$;*
2) *$i < j$ or $s < t$, and for $x \in \{i, j\}$, $y \in \{s, t\}$, $\alpha_{xy} < \alpha_{ij} \cdot \frac{\beta_{xy}}{\beta_{ij}} - 1$.*

Proof: See Appendix. ■

Lemma 1: Given E_{ij} and E_{st} that satisfy the condition of Theorem 1. Any pair of edges $e_1 \in E_{ij}$ and $e_2 \in E_{st}$ will reduce p_{ij} by the same amount.

Proof: As indicated in the proof of Theorem 1, which is independent of the choice of e_1 and e_2. ■

Intuitively, Theorem 1 guarantees that the swap of the appropriate pair of edges will always reduce the maximum linking probability p_{ij} and will not cause other linking probabilities to become larger than p_{ij}. Lemma 1 indicates that as long as the appropriate EECs are determined, the choice of edges within these EECs does not make any difference provided a valid swap can be made.

B. Degree-based Edge Deletion

Algorithm 2 takes as input a graph and a confidence threshold, and returns a τ-confident anonymous graph using edge deletion.

To preserve utility, Algorithm 2 also focuses on deleting edges of the leading EEC. To select an edge to delete, it estimates the amount of decrease of the maximum linking probability and the amount of increase of other linking probabilities that will be resulted from the deletion of the edge. The edge that will be selected should result in the largest *reduction to the maximum linking probability* (RMLP). If more than one such edge exists, the one that results in the smallest *increase of other linking probabilities* (IOLP) should be selected. Notice that it is possible that the deletion of the selected edge does not immediately improve the graph confidence. However, the algorithm will not stop if this situation occurs because by deleting more edges the maximum linking probability will eventually be reduced, and a τ-confident graph will be obtained. Furthermore, this can happen independent of the order in which edges are deleted.

To efficiently estimate the reduction of the maximum linking probability and the increase of other linking probabilities resulted from deleting an edge in the leading EEC, the algorithm does the following.

Suppose that the leading EEC is E_{ij} and the edge (u, v) is under consideration, where $u \in C_i$, $v \in C_j$, and VEC C_i contains vertexes of degree i. If (u, v) is deleted, u and v will be moved into VECs C_{i-1} and C_{j-1}, respectively. If $i \neq j$, the deletion of the edge will decrease the sizes of C_i and C_j and increase the sizes of C_{i-1} and C_{j-1}, each by one. If $i = j$, the deletion of the edge will decrease the size of C_i and increase the size of C_{i-1}, each by two. As u and v are moved, so will edges incident to u or v. To be specific, consider an edge (u, w), where w is in some VEC C_s. Once u is moved, this edge will also be moved from EEC E_{is} into EEC $E_{(i-1)s}$. This move will decrease the size of E_{is} and increase the size of $E_{(i-1)s}$, each by one. The size changes of EECs affected by moving v can be determined similarly. Thus, the size changes of VECs and EECs affected by the edge deletion can be efficiently determined without actually moving vertexes and edges. These size changes can then be used to calculate RMLP and IOLP.

Algorithm 3 is an alternative edge deletion method, which chooses a random edge, instead of the best edge, from the leading EEC for deletion.

V. Experiments

In this section, we present results of our empirical study. We implemented in Java the three algorithms described in Section IV and used *prefuse* (http://prefuse.org/), an open source graph package, for graph maintenance as well as for graph operations, such as edge deletion and edge swap.

We performed experiments on three datasets. In addition to EPINION and COA datasets, we also used KDDCUP[8] dataset, which is a collection of abstracts of papers in high energy physics published between 1992 through 2003. We extracted from the data a graph, which has a vertex for each

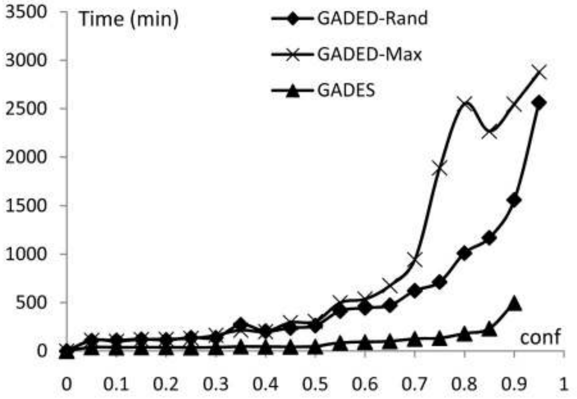


Fig. 3: Performance of Algorithms on EPINION Dataset

unique author and an undirected edge for each unique pair of authors who coauthored a paper. The graph contains 19,101 vertexes and 30,010 edges.

Our experiments were performed on a PC with 2.13GHz Pentium Core 2 processor and 2GB memory running a Ubuntu Linux operation system.

A. *Performance of GADES and GADED*

In this experiment, we compare the performance of the three algorithms. For COA and KDDCUP graphs, the execution of the algorithms is fast because only a small number of edges need to be changed. Here we only show the results obtained from the EPINION graph. As shown in Figure 3, edge swap (GADES) is always more efficient than edge deletion (GADED) especially for higher confidence thresholds. This is perhaps because that edge swap can always reduce the maximum linking probability but edge deletion may not. It is interesting that the two versions of edge deletion have almost identical performances when $\tau \leq 0.5$. But as confidence threshold becomes higher, GADED-Rand becomes more efficient than GADED-Max, which is as expected.

B. *Utility of Anonymous Graphs*

In this experiment, we compare the utility of anonymous graphs obtained by the three algorithms from the three datasets. Previous studies typically use one graph measurement as the utility measure. But since different graph applications can be sensitive to different graph measurements, we decide to measure the utility of anonymous graphs using three different measurements: namely, the changes of the graph edges, the degree distribution, and the clustering coefficients.

To measure the change of the graph edges, we count the number of edges that are deleted (due to edge deletion or edge swap) and added (due to edge swap), and calculate the relative ratio of edge change $RREC = \frac{|E|-|E'\cap E|}{|E|}$, where E is the set of edges in the original graph and E' is the set of edges in the anonymous graph. This ratio has been used to measure utility of anonymous graphs by several researchers[18], [11].

To measure the change of degree distributions, we calculate the distance between degree histograms of the original and the anonymous graphs. There are several distance measures for histograms. In our experiments, we use the earth mover distance (EMD) [13], which is the minimal cost of transforming one distribution into another. EMD has been used by researchers in many different areas [13].

The clustering coefficient (CC) of a vertex u is defined as $C_u = \frac{2l_u}{k_u(k_u-1)}$, where k_u is the number of neighbors of vertex u and l_u is the number of edges among neighbors of vertex u. Intuitively, clustering coefficient measures the closeness of the vertex and its neighbors in the graph and determines whether the graph has the small world characteristics. To measure the change of clustering coefficients, we calculate for each vertex u the difference of clustering coefficients $\Delta C_u = |C_u - C_u^{'}|$, where C_u and $C_u^{'}$ are calculated from the original and the anonymous graphs, respectively. We use the mean $MDCC = \frac{1}{|V|}\sum_{u\in V}\Delta C_u$ and the standard deviation $SDDCC = (\frac{1}{|V|-1}\sum_{u\in V}(\Delta C_u - MDCC)^2)^{\frac{1}{2}}$ of theses differences as a utility measure.

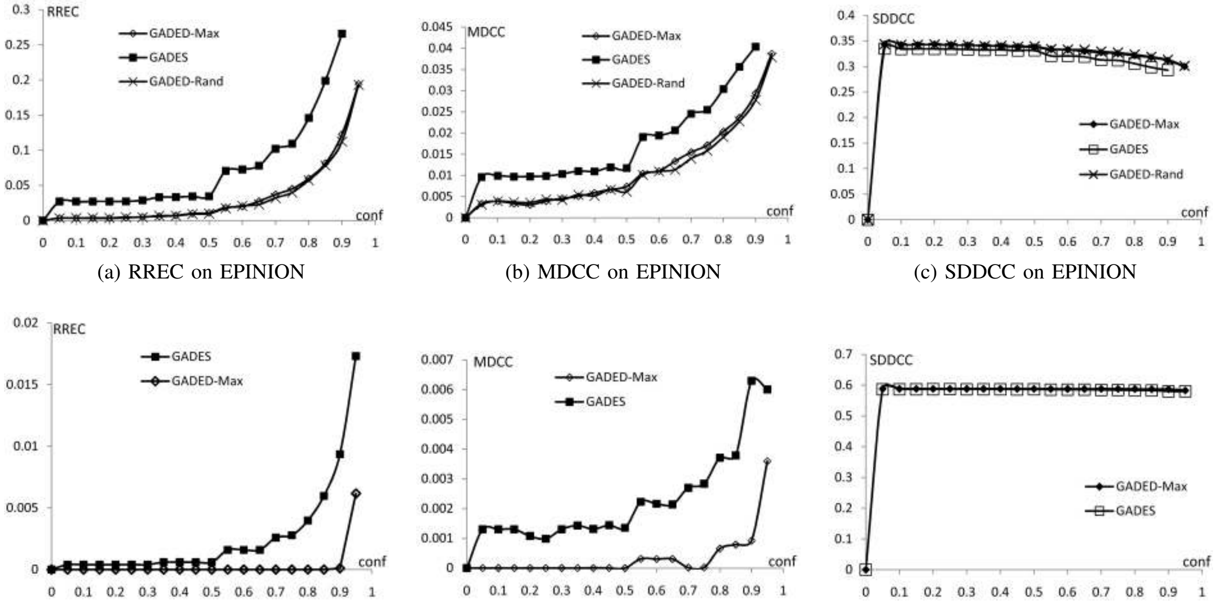
To obtain more reliable results, we repeat each experiment 5 times and report here the average results.

Figure 4 shows utility in terms of edge changes and clustering coefficient. In these figures, the X-axis is the graph confidence threshold, which ranges from 0 to 1, in increment of 0.1. The Y-axis is the corresponding utility measurements, also ranges from 0 to 1, where 0 indicates the highest and 1 indicates the lowest utility.

Figures 4d and 4a show the RREC of COA and EPINION, respectively. In both figures, edge deletion outperforms edge swap. However, for COA, the difference between edge swap and edge deletion are small for $\tau \leq 0.5$. We omitted the GADED-Rand in this figure because it is very similar to GADED-Max. For EPINION, the difference between edge swap and edge deletion is much bigger even for small τ. It also clearly shows that GADED-Max and GADED-Rand have almost identical performance. We also run the experiment on KDDCUP graph. Since the results are similar to those of COA, we do not show them here.

The remaining figures in Figure 4 show the results on clustering coefficient. They show that edge swap causes more changes of clustering coefficient than edge deletion does, and the mean change of clustering coefficient on EPINION graph is (about one order of magnitude) larger than on COA graph. The latter is expected because EPINION is denser, therefore, the change of an edge may change the cluster coefficients of more vertexes in EPINION than in COA. In terms of standard deviation, edge swap and edge deletion are not much different on each dataset, but the standard deviation on COA is about twice as large as on EPINION, indicating a wider range of changes of clustering coefficient. It is yet to determine how such differences impact various graph applications.

Figure 5 shows the result on degree histograms. Since edge swap never changes vertex degree, it has no impact on degree histogram, therefore, is not shown in the figure. For edge deletion, the results depend on datasets. For COA, the EMD is not noticeable for $\tau \leq 0.85$ and increased to 0.0002 for higher threshold. For EPINION, EMD starts to increase at $\tau = 0.6$ and increases more as the threshold becomes higher. But the largest EMD is still less than 0.00035. Thus for these two



(a) RREC on EPINION

(b) MDCC on EPINION

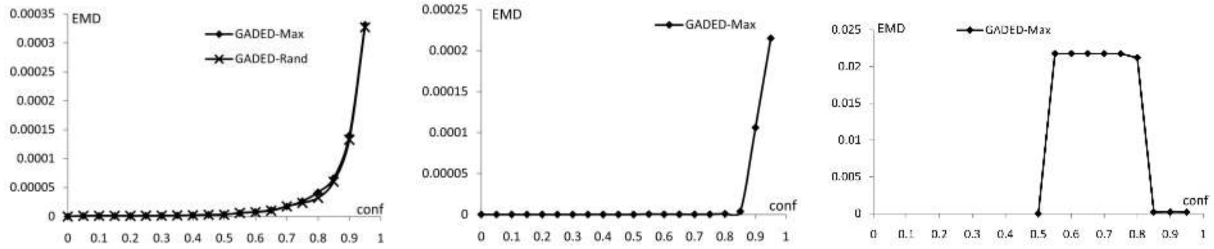
(c) SDDCC on EPINION

(d) RREC of COA

(e) MDCC of COA

(f) SDDCC of COA

Fig. 4: RREC, MDCC and SDDCC of Anonymous Graphs



(a) EMD on EPINION

(b) EMD on COA

(c) EMD on KDDCUP

Fig. 5: Earth Move Distance of Anonymous Graphs

datasets, the algorithms seem to preserve the original degree distributions. On the other hand, the result of KDDCUP is quite different. For $0.5 \leq \tau \leq 0.8$, the EMD jumps to and stays at about 0.02, a much higher value as compared to the EMD of COA and EPINION. This suggests that the effect on degree distributions is data-dependent.

To summarize, edge swap performs better than edge deletion. As for utility, edge swap performs worse than edge deletion on edge changes and clustering coefficient, but better than edge deletion on maintaining degree distribution. The two versions of edge deletion have comparable performance with GADED-Rand being slightly better on higher confidence threshold. This is somewhat unexpected because one would expect that GADED-Max will delete less number of edges and therefore perform better. We emphasize that these results are not conclusive, further study is required to better understand the performances.

VI. Conclusions

In this paper, we consider the edge anonymity in social networks. We presented a privacy notion of edge anonymity, the graph confidence, which captures privacy breach of an adversary who is able to use a type of VDT to pinpoint VECs of target persons. The notion is general enough to allow any VDT. We also consider a special type of edge anonymity problem where vertex degree is used as a VDT. We show that in some real-world social network graphs, especially those dense ones, edge disclosure can occur even if it is vertex k-anonymous. We present three heuristic algorithms to obtain τ-confident anonymous graphs. Our experiments, based on three real-world social networks and several utility measures, show that these algorithms can effectively protect edge anonymity and can produce anonymous graphs that have acceptable utility.

Acknowledgements

This material is based on work supported by National Science Fundation under Grant No. IIS-0524612. The authors would like to thank anonymous reviewers for their constructive comments.

References

[1] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *International Conference on Knowledge Discovery and Data Mining*, 2008.

[2] Reid Andersen, Christian Borgs, Jennifer Chayes, and Uriel Feige. Trust-based recommendation systems: An axiomatic approach. In *International World Wide Web Conference*, 2008.

[3] Lars Backstorm, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In *International World Wide Web Conference*, 2007.

[4] COA-Dataset. http://www.cs.helsinki.fi/u/tsaparas/MACN2006/data-code.html.

[5] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242, 2007.

[6] Epinions-Dataset. http://www.trustlet.org/wiki/Downloaded_Epinions_dataset.

[7] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. In *International Conference on Very Large Data Bases*, 2008.

[8] KDDCUP-Dataset. http://www.cs.cornell.edu/projects/kddcup/datasets.html.

[9] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web*, 1, 2007.

[10] Kun Liu, Kamalika Das, Tyrone Grandison, and Hillol Kargupta. Privacy-preserving data analysis on graphs and social networks. In Hillol Kargupta, Jiawei Han, Philip Yu, Rajeev Motwani, and Vipin Kumar, editors, *Next Generation of Data Mining*. CRC Press, 2008.

[11] Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In *ACM SIGMOD International Conference on Management of Data*, pages 93–106, 2008.

[12] Vibhor Rastogi, Michael Hay, Gerome Miklau, and Dan Suciu. Relationship privacy: Output perturbation for queries with joins. In *ACM Symposium on Principles of Database Systems*, 2009.

[13] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 2:99–121, 2000.

[14] Lisa Singh and Justin Zhan. Measuring topological anonymity in social networks. In *IEEE International Conference on Granular Computing*, 2007.

[15] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 2nd edition, 2001.

[16] Xiaowei Ying and Xintao Wu. Randomizing social networks: a spectrum preserving approach. In *SIAM International Conference on Data Mining*, 2008.

[17] Elena Zheleva and Lise Getoor. Preserving the privacy of sensitive relationships in graph data. In *ACM International Workshop on Privacy, Security, and Trust in KDD (PinKDD)*, 2007.

[18] Bin Zhou and Jian Pei. Preserving privacy in social networks against neighborhood attacks. In *IEEE International Conference on Data Engineering*, 2008.

[19] Bin Zhou, Jian Pei, and Wo-Shun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explorations*, 10:12–22, 2008.

Appendix

Proof: (if) Assume that i, j, s, t contain at least two distinct values and each of these values appears at most twice, and one of the two conditions holds.

If condition 1 holds, $e_1 \in E_{ii}$, $e_2 \in E_{ss}$, and E_{ii} and E_{ss} must be different EECs. The edge swap will delete one edge from each of E_{ii} and E_{ss}, and add two edges to E_{is}. Since $\alpha_{is} < \alpha_{ii} \cdot \frac{\beta_{is}}{\beta_{ii}} - 2$, by Definition 4, $p_{ij} = p_{ii} > p^{'}_{is}$. Since E_{is} is the only EEC receives a new edge, we are done.

If condition 2 holds, we consider three cases.

Case 1: $i = j$ and $s < t$. In this case, $e_1 \in E_{ii}$, $e_2 \in E_{st}$, and since i, j, s, t contain at least two distinct values each of which appears at most twice, E_{ii} and E_{st} must share no VEC. Thus, after the edge swap, EECs E_{is} and E_{it} each receives one new edge. Since for $x \in \{i, j\}$, $y \in \{s, t\}$, $\alpha_{xy} < \alpha_{ij} \cdot \frac{\beta_{xy}}{\beta_{ij}} - 1$, by Definition 4, we have $p_{ij} = p_{ii} > p^{'}_{is}$ and $p_{ij} = p_{ii} > p^{'}_{it}$.

Case 2: $i < j$ and $s = t$. In this case, $e_1 \in E_{ij}$, $e_2 \in E_{ss}$. Similarly, E_{ij} and E_{ss} must share no VEC. Thus, after the edge swap, EECs E_{is} and E_{js} each receives one new edge. Since for $x \in \{i, j\}$, $y \in \{s, t\}$, $\alpha_{xy} < \alpha_{ij} \cdot \frac{\beta_{xy}}{\beta_{ij}} - 1$, by Definition 4, we have $p_{ij} > p^{'}_{is}$ and $p_{ij} > p^{'}_{js}$.

Case 3: $i < j$ and $s < t$. In this case, $e_1 \in E_{ij}$ and $e_2 \in E_{st}$, and after the edge swap, EECs that receive new edges will be depending on the order of i, j, s, t. These orderings are $s < t \leq i < j$, $s < i \leq t < j$, $s \leq i < j \leq t$, $i < s < t < j$, and $i < j \leq s < t$. The proofs are similar for these orderings. We only show the proof for $i < j = s < t$. For this ordering, E_{it} and E_{js} each receives a new edge. Since for $x \in \{i, j\}$, $y \in \{s, t\}$, $\alpha_{xy} < \alpha_{ij} \cdot \frac{\beta_{xy}}{\beta_{ij}} - 1$, by Definition 4, we have $p_{ij} > p^{'}_{it}$ and $p_{ij} > p^{'}_{js}$.

(only if) Assume that for each EEC E_{xy} in G' that received a new edge, the corresponding linking probability $p^{'}_{xy}$ is less than p_{ij}.

By Definition 4, $p_{ij} > p^{'}_{xy}$ implies $\alpha_{xy} < \alpha_{ij} \cdot \frac{\beta_{xy}}{\beta_{ij}} - c$, where $c = 1$ or 2 depending on the number of new edges E_{xy} receives. Notice that, since the edge swap can at most involve four VECs, we have $x, y \in \{i, j, s, t\}$.

We prove that indexes i, j, s, t contain at least two distinct values and each of these values appears at most twice. This can be proved by contradiction. Assume that there is only one distinct value. Then $i = j = s = t$. That is, $e_1, e_2 \in E_{ii}$. The swap will remove two edges from E_{ii} and add two new edges back to E_{ii}. Therefore, $p_{ij} = p_{ii} = p^{'}_{ii} = p^{'}_{xy}$, a contradiction. All cases in which i, j, s, t contain two distinct values and one value appears three times can be proved similarly.

Thus, indexes i, j, s, t contain at least two distinct values and each of these values appears at most twice. There are 5 cases in which i, j, s, t contains exactly two pairs of identical indexes, where the first pair is: $i = j$, $i = s$, $i = t$, $j = s$, or $j = t$. If $i = j$ then $s = t$. Therefore, $e_1 \in E_{ii}$ and $e_2 \in E_{ss}$. After the edge swap, EEC E_{is} will receive two edges. Thus, $p_{ij} > p^{'}_{xy}$ implies $\alpha_{is} < \alpha_{ii} \cdot \frac{\beta_{is}}{\beta_{ii}} - 2$. This givens the condition 1.

The remaining four cases imply that $i < j$ and $s < t$. The proofs of these cases are similar. We show the proof for the case $i = s$. In this case, $j = t$. Therefore, $e_1, e_2 \in E_{ij}$ and after the edge swap, E_{ii} and E_{jj} each receives one new edge. Thus, $p_{ij} > p^{'}_{xy}$ implies $\alpha_{ii} < \alpha_{ij} \cdot \frac{\beta_{ii}}{\beta_{ij}} - 1$ and $\alpha_{jj} < \alpha_{ij} \cdot \frac{\beta_{jj}}{\beta_{ij}} - 1$.

The cases in which $i, j, s., t$ contain exactly three and exactly four distinct values can be proved similarly. Notice that in these cases, we have either $i < j$ or $s < t$, and the EECs that can ever receive a new edge is among E_{is}, E_{it}, E_{js}, and E_{jt} (notice that $E_{xy} = E_{yx}$). The details are omitted. ■