A Guide to Differential Privacy Theory in Social Network Analysis

Christine Task, Chris Clifton
Department of Computer Science
Purdue University
West Lafayette, IN
Email: ctask@purdue.edu, clifton@purdue.edu

***Abstract*—Privacy of social network data is a growing concern which threatens to limit access to this valuable data source. Analysis of the graph structure of social networks can provide valuable information for revenue generation and social science research, but unfortunately, ensuring this analysis does not violate individual privacy is difficult. Simply anonymizing graphs or even releasing only aggregate results of analysis may not provide sufficient protection. Differential privacy is an alternative privacy model, popular in data-mining over tabular data, which uses noise to obscure individuals' contributions to aggregate results and offers a very strong mathematical guarantee that individuals' presence in the data-set is hidden. Analyses that were previously vulnerable to identification of individuals and extraction of private data may be safely released under differential-privacy guarantees. We review two existing standards for adapting differential privacy to network data and analyse the feasibility of several common social-network analysis techniques under these standards. Additionally, we propose *out-link privacy*, a novel standard for differential privacy over network data, and introduce two powerful out-link private algorithms for common network analysis techniques that were infeasible to privatize under previous differential privacy standards.**

I. Introduction

Social networks are powerful abstractions of individuals and the relationships that connect them; social network analysis can be a very powerful tool. For example, understanding how well-connected a network is can aid in the development of word-of-mouth marketing campaign: How quickly will word of a product spread? Similar analysis is useful in epidemiology, predicting spread of a disease.

However, data about people and their relationships is potentially sensitive and must be treated with care to preserve privacy. Generally, social network graphs are anonymized before being made available for analysis. However, as several recent incidents have demonstrated, releasing even anonymized graphs may lead to re-identification of individuals within the network and disclosure of confidential information, with serious consequences for those involved. In 2007, Netflix released the Netflix Prize data-set, containing anonymized data about the viewing habits of its members, for public analysis by information retrieval researchers. Within a year, it had been demonstrated that wide-spread de-anonymization of individuals in the data-set was possible using public information from the Internet Movie Database [1]. By 2009, Netflix was involved in a lawsuit with one of its members who had been victimized by the resulting privacy invasion.

Privacy researchers have attempted to improve the security provided by graph anonymization techniques by adding noise to the node parameters and structure of the graph [2]. However, even a noisy graph structure with no node parameters whatsoever can be subject to deanonymization, particularly if an attacker has background knowledge of the network data [3]. For example, knowing the friendship relationships of a few individuals can make them identifiable in the released graph, leading to identification of their friends (and disclosure of information, such as other relationships, that those friends might not want publicly revealed.) As global social networks become more broadly accessible, these types of background knowledge are more readily available [3].

Differential privacy is a privacy standard developed for use on tabular data that provides strong guarantees of privacy without making assumptions about an attacker's background knowledge [4]. Differentially-private queries inject randomized noise into query results to hide the impact of adding or removing an arbitrary individual from the data-set. Thus, an attacker with an arbitrarily high level of background knowledge cannot, with a high degree of probability, glean any new knowledge about individuals from differentially-privatized results; in fact, the attacker cannot guess whether any given individual is present in the data at all.

While many of the privacy concerns associated with social-network analysis could be relieved by applying differential-privacy guarantees to common social-network analysis techniques, researchers have struggled to develop suitable adaptations of these techniques. Two principal difficulties arise: The adaptation of differential privacy from tabular data to network data, and the high sensitivity of social-network metrics to relatively small changes in the network structure.

In this paper, we present a practical introduction to the application of differential privacy to social networks. We provide the following:

- A straightforward introduction to traditional differential privacy;
- A discussion of two known differential-privacy standards for network data, as well as the contribution of a new third standard, *out-link privacy*, which provides strong privacy guarantees with the introduction of very small noise;
- A study of the feasibility of common social-network

978-0-7695-4799-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ASONAM.2012.73

analysis techniques under differential-privacy;

- The contribution of two new algorithms, satisfying *out-link privacy* that use ego-network style analysis to provide approximate results for queries that are too sensitive to perform under previous standards.

II. Traditional Differential Privacy

Differential privacy was developed by Cynthia Dwork at Microsoft Research Labs [4]. It does not define a specific technique or algorithm; instead it states a mathematical guarantee of privacy that sufficiently well-privatized queries can satisfy. Consider a common sequence of events in social science research: a survey is distributed to individuals within a population; a subset of the population chooses to participate in the survey; individual information from the surveys is compiled into a data-set and some analysis is computed over it; the analysis may be privatized by the injection of random noise; and the final privatized result is released to the general public. Differentially-private queries offer a rigorous mathematical guarantee to survey participants that the released results will not reveal their participation in the survey.

We first introduce a few useful notations: I is set of individuals who contribute information to the data-set D_I (e.g., survey participants). The set of *all possible* data-sets is $\mathcal{D}$. We use $F : \mathcal{D} \rightarrow \Re^k$: to refer to the desired non-privatized analysis performed on a data-set and $Q : \mathcal{D} \rightarrow \Re^k$ to refer to the privatized implementation of F. We refer to the publicly released, privatized analysis results as R.

If R are the privatized query results that are released to the public, then R is the only evidence an attacker has about the nature of D_I. We introduce a possible-worlds model to understand how differential privacy works. We define D_I to be *true world* from which the analysis was taken. We also define any data-set that differs by the presence or absence of one individual to be a "neighboring" possible world: thus D_{I-Bob} is the neighboring possible world of D_I in which Bob chose to not participate in the survey.

We require that an attacker possessing the privatized results R be unable to determine whether or not Bob (or any other specific individual) took the survey, i.e. whether or not R are the results from an analysis of D_I or D_{I-Bob} (or, indeed, any neighboring world of D_I). Therefore, R should be a plausible result from any neighboring world of D_I.

Formally, D_I neighbors D_J iff $D_I = D_{J \pm x}$ for any x in the population, and:

Definition 1: A randomized query

$$Q : \mathcal{D} \rightarrow \Re^k$$

satisfies *ϵ-differential privacy*[4] if, for *any* two possible neighboring data-sets D_1, D_2 and *any* possible query result R:

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

Here ϵ is a small, positive value that controls the trade-off between privacy and accuracy, and is chosen by the person administering the privacy policy. To make the definition more intuitive, consider that if we set $\epsilon = ln(2)$, the above states that the result R is at most twice as likely to be produced by the true world as by any of its neighbors. Setting a smaller ϵ will provide greater privacy at the cost of additional noise, as we will demonstrate below.

The difference between the results from the true world D_1 and its neighbor D_2 is the difference the privatization noise will need to obfuscate in order for the privatized results to not give evidence about whether D_1 or D_2 is the true world. The upper bound of this difference over $D_I \in \mathcal{D}$ is the *sensitivity* of query F.

Definition 1: The *global sensitivity* of a function $F : \mathcal{D} \rightarrow R^k = A$ is [1]:

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

over all pairs of neighbouring data-sets D_1, D_2.

Intuitively, the sensitivity of a query is the *greatest* possible impact that adding or removing an arbitrary individual from the data-set can have on the query results, over *any* possible data-set. Suppose our analysis F asks two questions: "How many people in I are depressed?" and "How many people in I have fewer than 3 friends?" Then both answers can change by at most 1 when a single individual is added to or removed from I, and ΔF = 2. If our analysis instead asks: "How many people in I are depressed?" and "How many people in I are happy?" then at most *one* answer can change by at most 1, and $\Delta F = 1$. Note that histograms, which partition the individuals of the data set into 'bucket' counts, have a sensitivity of 1: removing or adding an individual will change at most one bucket count by at most 1. This very low sensitivity makes histograms a useful tool in differentially private data-mining [4], [5], [6].

We can create a differentially private query Q by adding noise to F that is calibrated to cover up ΔF [4]:

Theorem 1: If $F : \mathcal{D} \rightarrow \Re^k$ is a $k - ary$ function with sensitivity ΔF then the function $F(D) + Lap^k(\Delta F/\epsilon)$ is ϵ-differentially private, where $Lap^k(\lambda)$ is a k-tuple of values sampled from a Laplacian random variable with standard deviation $\sqrt{2}\lambda$.

The standard deviation of the Laplacian noise values is $\sqrt{2}\Delta F/\epsilon$. Thus the noise will be large if the function is very sensitive, or if ϵ is small. If we set $\epsilon = ln(2)$ on a query with sensitivity $\Delta F = 2$, the standard deviation of our added noise will be close to 4.

It's important to note that ΔF is an upper bound taken across *all possible* pairs of neighboring data-sets; it is independent of the true world. Intuitively, this is necessary because noise values that are dependent on the nature of the true world may introduce a privacy leak themselves. For example, when querying the diameter of a social network, if Alice forms the only bridge between otherwise unconnected subgraphs in the true world, removing her from the data-set causes a difference of ∞ in the graph diameter. Noise values calibrated to this true world must be arbitrarily large (and, in fact, will obliterate the

[1] The L_1-norm of $x \in \Re^n$ is defined as $\|x\|_1 = \Sigma_{i=1}^n |x_i|$.

utility of the result). However, consider a neighboring *possible* world including Bob, who forms a second bridge between the subgraphs; if this possible world were the true world, the difference in diameter caused by adding or removing a node would be finite, and if we calibrated the noise to that difference, it would be relatively small. If we chose our noise values based on the true world, an attacker could easily determine whether or not Bob was in the network: a result of $R = 300,453.23$ would imply Bob was absent, while the result $R = 4.23$ would indicate that Bob was present. To prevent this, global sensitivity is based on the worst-case scenario for the query. In this case, this implies that diameter is a query too sensitive to be feasibly privatized.

Techniques exist that do not use the global sensitivity upper bound, such as privatization algorithms using *smooth sensitivity*, and these have been applied successfully to graph analysis problems. However, these techniques satisfy a weaker definition of differential privacy, and in some cases computing how much noise is required to privatize a given D_I may be infeasible. We will focus on techniques that satisfy strict ϵ-differential privacy in this paper, but we recommend looking at [7] for more information on alternative approaches.

III. Differential Privacy and Network Data

The above definition for differential privacy assumes all information about a data-set participant is provided by the participant themselves; protecting an individual's presence in the data-set then protects all the information regarding them. The situation changes when we ask survey participants to provide information about other individuals.

We will refer to individuals who contribute their knowledge to the data-set as *participants*, and individuals who have information provided *about* themselves (by others) as *subjects*. Traditional differential privacy protects participants only, and in many cases it seems clear that subject privacy is unnecessary: if a survey counts the students who attended the "Coffee with the Dean" event, the dean's privacy is not important. By contrast, a study that counts students who report having sexual relations with the football captain exposes extremely sensitive information about its subject. Social networks are often collected from populations of interest by having participants list the full names of their friends within the population; these relationships form directed network edges leading from the participant's node to the nodes of each of their friends [8]. In this case, the friends are subjects of the participant's survey data, but the participant herself may also be the subject of some of her friends' survey data (if they also submit surveys). This presents a complex situation in which to apply differential privacy.

The core of the differential privacy guarantee is that the privatized result R is difficult to attribute to the true world vs. one of its neighboring possible worlds. Adapting differential privacy to networked data amounts to deciding what we mean by "neighboring worlds" in this context. There are several possibilities; each one provides a different level of privacy guarantee and deals with a different type of "gap" between worlds. As always, there is a trade-off between privacy and utility: in general, the stronger the privacy guarantee, the more noise will be required to achieve it. We will describe two network privacy standards, *node privacy* and *edge privacy*, which have appeared in the literature.

Additionally, we propose a novel third standard, *out-link privacy*, which requires less noise than existing standards; gives a reasonably strong guarantee of privacy similar to traditional differential privacy; and enables certain queries that required levels of noise that rendered results meaningless under existing standards.

A. Node Privacy

A privatized query Q satisfies *node-privacy* if it satisfies differential privacy for all pairs of graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ where $V_2 = V_1 - x$ and $E_2 = E_1 - \{(v_1, v_2) | v_1 = x \vee v_2 = x\}$ for some $x \in V_1$

In node privacy, if the true world is a given social network G, the neighboring possible worlds are ones in which an arbitrary node, and *all* edges connected to it, are removed from or added to G. This privacy guarantee completely protects *all* individuals, both participants and subjects. An attacker in possession of R will not be able to determine whether a person x appears in the population at all. This places *extremely* severe restrictions on the queries we are able to compute, as we will demonstrate in section IV, and in many cases, node-privacy may be an unnecessarily strong guarantee.

B. Edge Privacy

A privatized query Q satisfies *edge-privacy* if it satisfies differential privacy for all pairs of graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ where $V_1 = V_2$ and $E_2 = E_1 - E_x$ where $|E_x| = k$

In edge privacy, if the true world is the social network G, neighboring possible worlds are ones in which k arbitrary edges are added or removed from G. An attacker in possession of R won't be able to determine with high certainty whether individuals x and y are friends, and an individual node in the graph can plausibly deny the existence of up to k of its friendships with other nodes. Single edge privacy, with $k = 1$, is the standard most often used in existing literature on differentially private graph analysis. This is a weaker guarantee than node-privacy: high-degree nodes may still have an identifiable effect on query results, even though their individual relationships are protected. However, this is a sufficiently strong for many applications, and enables many more types of queries to be privatized than the severely-restrictive node-privacy.

C. Out-link Privacy

A privatized query Q satisfies *out-link privacy* if it satisfies differential privacy for all pairs of graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ where $V_1 = V_2$ and $E_2 = E_1 - \{(v_1, v_2) | v_1 = x\}$ for some $x \in V_1$.

This privacy guarantee protects the data contributed by data-set *participants*, using the same conceptual privacy standard as the original definition of differential privacy. Given that the true world is a social network G, the neighboring possible worlds are ones in which an arbitrary node and all of its *out-links* are removed from or added to G. An attacker in possession of R won't be able to determine whether a person x supplied their data (submitted a survey) to help produce the graph. This privacy guarantee is strictly weaker than node privacy, but compares well with single edge privacy for many queries. Any participant can plausibly deny its out-links, or, equivalently, any participant can plausibly deny one in-link from another participant node. Analogous to k-edge privacy, we can also provide k-out-link privacy by considering neighboring worlds that differ from the true world by the out-links of up to k nodes. Note that 2-out-link privacy allows two nodes to *simultaneously* deny all out-links, and as a result, this enables a complete mutual edge to be protected (providing single-edge privacy in addition to out-link privacy). In general, a k-level privacy guarantee can be satisfied by scaling the added noise by k.

Out-link privacy improves on edge-privacy by reducing the distinctive signature of high-degree nodes in the data-results, through protecting all relationships cited *by* the popular person: although others may still claim to be friends with her, she can plausibly deny those relationships are mutual. Additionally this standard simplifies sensitivity computation and noise addition, enabling many queries that would be unfeasible under both node and edge privacy as we will demonstrate in section IV.

Below we will discuss the application of these privacy standards to common social network analysis tasks such as triangle counts (and subgraph-counts generally), degree distributions, centrality measures, graph-modeling, and other differentially privatized network analyses from the existing literature. In addition to covering previous work, we provide several infeasibility proofs and propose two original algorithms applying out-link privacy to common problems in social network anlaysis.

IV. Applications of Differential Privacy to Social Network Analysis

We now present a straightforwards guide to the application of differential privacy to several common social network analysis techniques.

A. Triangle Counting

Triangles, instances in which two of an individual's friends are themselves mutual friends, indicate social cohesion in the network. Triangle counts are the key parameter in the clustering coefficient, a common metric for describing and comparing graphs. Similarly, counts of other subgraphs such as stars, or squares, are used as graph statistics for graph similarity comparisons [9], [10]. All subgraph counts have similar privacy properties to the triangle count privatization described below.

Differentially private triangle counts are not feasible under simple node-privacy. In the worst case, adding a node to a complete graph of size n (a graph containing all possible edges), will introduce $\binom{n}{2}$ new triangles (Figure 1). Since the change is dependent on the size of the graph, the global sensitivity of the query in general is unbounded: it's impossible to compute a finite global upper-bound (see Section III).

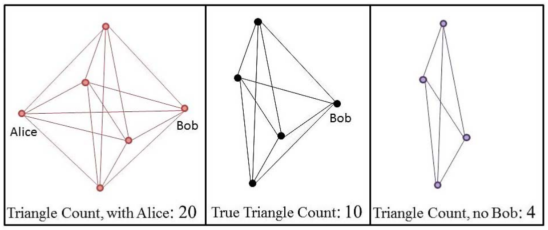


Fig. 1. Node-sensitivity of triangle-counts is a function of n, and thus is unbounded in general.

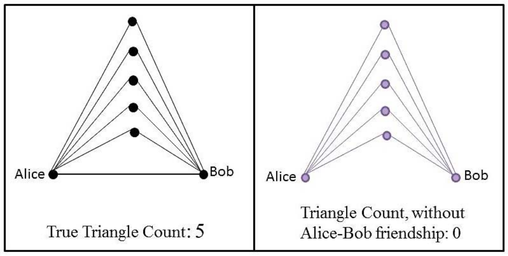
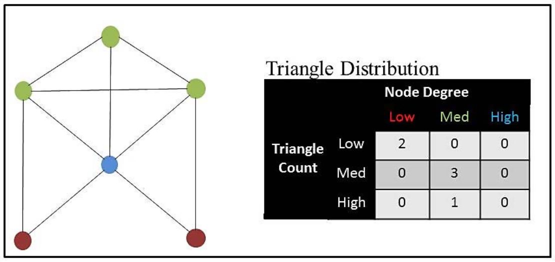


Fig. 2. Edge-sensitivity of triangle-counts is a function of n, and thus is unbounded in general.

For similar reasons to node privacy, edge privacy is also not feasible for triangle-counts. In the worst case, removing an edge from a graph with n nodes can remove $n-2$ triangles (Figure 2). Since the sensitivity is a function of the graph size, it is unbounded in general.

We now propose a method for privatizing information about triangle counts and clustering coefficients under out-link privacy, using a somewhat modified version of the query that more closely mimics the information gathered from a real world social-network survey. To do this, we introduce a simple, powerful method that can be applied to gather private estimates of a variety of useful statistics over nodes in the graph.

By focusing on protecting the knowledge each individual has about their role with respect to the network, out-link privacy fits naturally with the techniques of *ego-network analysis*, an approach to social network analysis which focuses



Triangle Distribution

		Node Degree		
		Low	Med	High
Triangle Count	Low	2	0	0
	Med	0	3	0
	High	0	1	0

Fig. 3. The triangle distribution allows us to present clustering information with an out-link sensitivity of 1.

on the network as viewed by the individuals belonging to it [11]. In this approach, a network with n members is broken into n overlapping ego-network subgraphs, each consisting of a individual 'ego' node and his or her immediate neighborhood of friends (referred to as alters). A survey collecting information about the triangles in an individual's ego-network might look like Algorithm 1.

Algorithm 1 A survey gathering information about triangles.

```
function TRIANGLEQUERY
    $friendlist$ ← Query("Who are your friends?")
    $friendpairs$ ← CrossProduct($friendlist, friendlist$)
    $outdegree$ ← Size($friendlist$)

    $triangles$ ← Query("Which of these pairs are friends with each other?", $friendpairs$)
    $trianglecount$ ← Size($triangles$)
    return ($outdegree, trianglecount$)
end function
```

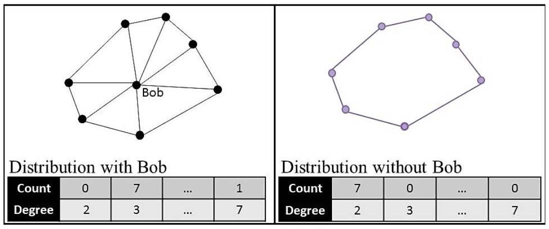
The only data that is retained by the researcher is, for each individual x: $outdegree(x)$, the number of friends the individual has, and $trianglecount(x)$, the number of triangles the individual participates in. These statistics are sufficient to determine the local clustering co-efficient of the node: the ratio between the number of triangles the node participates in and the maximum possible number of triangles for a node of that degree [9].

Out-degree and local clustering data from this survey can be collected into a two-dimensional histogram that provides detailed information about the patterns of social cohesion of the graph and has a very low sensitivity under out-link privacy: removing or adding an individual's survey data to the histogram only alters one partition count by at most one, and thus the noise required to privatize this data-structure would be very small. Histograms with fewer partitions and larger count values in each partition are less sensitive to added noise; we propose Algorithm 2 which produces a very flexible, robust, and safely privatized representation of the social cohesion patterns in the network using local triangle counts.

Algorithm 2 takes as input two node-degree threshold values, deg_{low}, deg_{med} and uses these to partition the $(outdegree, trianglecount)$ data-points collected from the survey into low, medium and high degree nodes. The algorithm then computes the local clustering coefficient of each node and further partitions nodes by these values, creating a histogram with nine partitions (see Figure 3). Laplacian noise sufficient to cover a function sensitivity of 1 is added to each partition, and the privatized result may be released. We can consider the effect of this noise in terms of how many of the noisy, privatized partition counts can be expected to differ measurably from their true values. With only nine counts and a sensitivity of 1, the expected number of privatized partition counts which will differ from their true values by more than 3, is less than 0.25. The released histogram accurately and succinctly captures useful information about the distribution of meaningful local patterns across the graph.

Algorithm 2 Privatizing local clustering coefficient distribution data.

```
function PRIVATECLUSTERING($deg_{low}, deg_{med}, data$)
    Initialize($bins[][]$)
    for all $(nodeDegree, triangleCount) \in data$ do
        $degBin$ ←Partition($nodeDegree, deg_{low}, deg_{med}$)
        $localCluster \leftarrow triangleCount/(nodeDegree * (nodeDegree - 1))$
        $triBin$ ←Partition($localCluster, 1/3, 2/3$)
        $bin[degBin][triBin] \leftarrow bin[degBin][triBin] + 1$
    end for
    for $i = 0 \rightarrow 2, j = 0 \rightarrow 2$ do
        $bins[i][j] \leftarrow bins[i][j] +$ LaplacianNoise(1)
    end for
    return $bins$
end function
```



Distribution with Bob

Count	0	7	...	1
Degree	2	3	...	7

Distribution without Bob

Count	7	0	...	0
Degree	2	3	...	7

Fig. 4. Node-sensitivity of degree distribution queries is a function of n, and thus is unbounded in general.

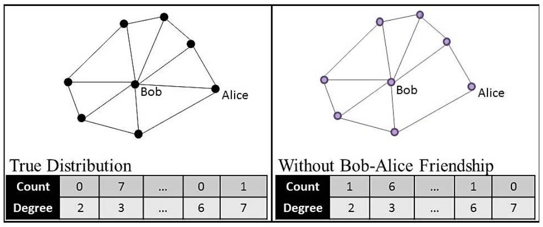
The same simple approach can be used to collect and privatize any information available within an ego-network, simply by restructuring the survey appropriately. For example, replacing question 2 in the survey of 1 by the question "For each of your friends, add a check mark if the two of you share at least one additional, mutual friend" will collect information about the probability that an edge participates in a triangle. The question "Are you part of a group of at least k friends who are all mutual friends with each other?" collects statistics about cliques in the graph.

In cases where pre-existing, undirected social network data must be privatized, the survey-collection approach described above may be simulated by considering each node's immediate neighborhood as their ego-network view, and sub-sampling by introducing α probability that the ego is unaware of any given edge between its alters.

B. Degree Distribution

The degree distribution of a graph is a histogram partitioning the nodes in the graph by their degree; it is often used to describe the underlying structure of social networks for purposes of developing graph models and making similarity comparisons between graphs [12].

Although degree distributions are represented as histograms, the sensitivity is not small under node privacy because one node affects multiple counts in the distribution: removing a



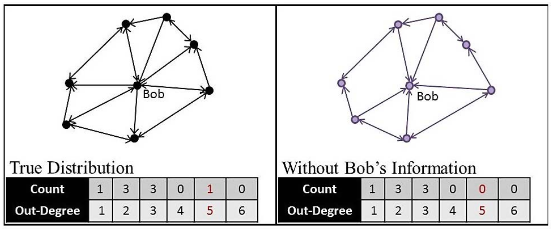
True Distribution

Count	0	7	...	0	1
Degree	2	3	...	6	7

Without Bob-Alice Friendship

Count	1	6	...	1	0
Degree	2	3	...	6	7

Fig. 5. Edge-sensitivity of degree distirubtion queries is 4: at most four values can change by one when a node is added or removed.



True Distribution

Count	1	3	3	0	1	0
Out-Degree	1	2	3	4	5	6

Without Bob's Information

Count	1	3	3	0	0	0
Out-Degree	1	2	3	4	5	6

Fig. 6. Out-degree sensitivity = 1. Protecting the out-edges of a node provides privacy with relatively little effect on the degree distribution.

node from the graph reduces the degree of all nodes connected to it. A node with k edges can affect a total of $2k + 1$ values of the distribution (Figure 4). In the worst case, adding a node of maximal degree will change $2n + 1$ values, and since this sensitivity is dependent on n, it will be unbounded in general (see Section III).

Edge privacy is feasible for degree distributions, however. Removing one edge from the graph changes the degree of two nodes, and affects at most four counts (Figure 5). Under k-edge privacy, the sensitivity is $4k$. With a sufficiently large graph, this is a negligable amount of noise, and the utility of this technique has been successfully demonstrated [6].

Out-link privacy, in contexts where it's deemed sufficient, requires even less noise for degree distributions. Here, we consider just the distribution of out-degrees, the result of asking participants, "How many friends do you have?" Removing one node and its out-links from the graph affects only one value in the degree distribution (Figure 5). Under this privacy standard, a high-degree node may still leave evidence of its presence in the data-set through the out-degrees of its friends. However, there are many possible explanations for a slightly higher-than-expected degree among nodes in the graph: they may represent additional friendships among the nodes, or outside friendships with individuals who were non-participants in the survey. Exploiting this vulnerability to guess the presence of a high-degree node with any certainty would require an attacker to possess near complete information about the true social network.

C. Centrality

Centrality measures attempt to gauge the relative "importance" of specific individuals within the social network; they may be studied on a per-node basis, identifying influential members of the community, or as distribution scores providing information about the overall behavior of the social network [13]. The simplest centrality measure is node degree: nodes with high degree are more likely to be influential in the network. However, other centrality measures take into account information from across the network: *betweeness* scores individuals by the number of shortest-paths between other pairs of nodes across the network that pass through them, and *closeness* scores nodes by the sum of their distances to all other nodes in the graph.

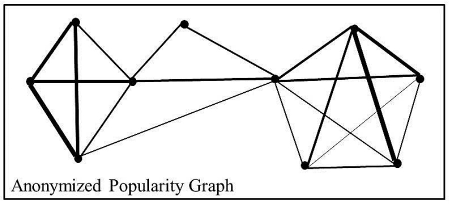


Fig. 7. A Popularity Graph with edge thickness indicating edge-weight

The two more complex centrality measures present difficulties for traditional approaches to differential privacy in social networks. Clearly, it is impossible to release a named list of influential individuals under node-privacy. But even distributions of centrality scores can be very sensitive, under both node and edge privacy, due to the role of bridges in the graph. Removing a node, or edge, which forms the only connection between two otherwise disconnected subgraphs will have a catastrophic affect on path distances in the network, causing finite distances to become infinite, and thus will drastically alter betweeness and closeness scores. In general, privatizing traditional centrality measures under traditional differential privacy remains an open problem.

We propose a very different approach for collecting and privatizing information about influential nodes within a network; one that satisfies out-link privacy (by protecting individuals' data contributions) and leverages individuals' knowledge about their community. We define a *popularity graph*: a synthetic network that represents the social structure among influential community members (Algorithm 3).

Individuals in the population are asked to "list up to three of your most popular friends within the specified population group". A base graph is created containing nodes for all members of the population group, and undirected edges of weight 0 are added between all pairs of nodes. The data collected from the survey is then added to the graph: when two popular people are listed on the same survey, the weight of the edge connecting them is incremented. For example, if a person submits a survey listing three popular friends, weights of every edge in the triangle connecting those friends will be incremented. The sensitivity of the popularity graph is 3, since a maximum of 3 edge-weight values can change if a participant adds or retracts their data.

To privatize the data, appropriate Laplacian noise to cover a function sensitivity of 3 is added to all edge-weights. Then

Algorithm 3 Privatizing centrality data.

```
function PRIVATECENTRALITY($importanceT, data_I$)
    $V \leftarrow population$
    $E[i][j] \leftarrow 0 \ \forall i, j \in V$
    for all $i \in I$ do
        $\forall p_j, p_k \in data_I[i],\ E[p_j, p_k] \leftarrow E[p_j, p_k] + 1$
    end for
    for all $i, j \in population$ do
        $E[i, j] \leftarrow E[i, j] +$ LaplacianNoise(3)
        if $E[i, j] < importanceT$ then
            $E[i, j] \leftarrow 0$
        end if
    end for
    return $PopularityGraph = (V, E)$
end function
```

two post-processing steps are applied: edges with low weight are eliminated, and the graph is anonymized. The resulting weighted popularity graph is published (Figure 7). This graph can be used to understand the underlying social influence structure of the population, identifying social clusters and the bridges between them. The privacy of data provided by the query participants is fully protected; however, the subjects who appear as nodes in the graph will clearly be less secure and this analysis may not be appropriate in all contexts. For many population though, the popularity graph should be sufficient protection: anonymity, noisy edges, and the fact that the artificially-constructed graph will lack detailed substructures often used for re-identification attacks, will all contribute to protecting the privacy of the query subjects.

D. Graph-modeling and Social Recommendations

Several groups have proposed differentially private approaches to creating graph models–randomized synthetic graphs that are generated to be similar to a true, private, social network and thus can be studied safely in place of the original graph. The Stochastic Kronecker graph model has been privatized under edge-privacy [14], and several other groups have developed their own models that satisfy differential edge privacy, [15], [16], [17].

We also note that the results from our proposed outlink privatized degree distribution and triangle statistics (see Sections IV-B and IV-A) could provide privatized input for the Transitive Chung Lu graph model proposed by [18]. This model is somewhat unique in the literature for its ability to generate graphs that match both the degree distribution and clustering coefficient of the original target graph.

Finally, the possibilities and difficulties of applying edge-privacy standards to social network recommendation systems are explored in [19].

V. CONCLUSIONS

Differential privacy represents a potentially powerful tool for analysing social networks while providing strong guarantees of privacy for individual participants. The application of differential-privacy guarantees to social-network analysis allows results to be released with confidence that individual data will not be compromised by malicious attackers, even with the benefit of arbitrary background knowledge.

By providing this guide to differentially private social network analysis, along with new, powerful techniques for privatizing social-network data, we hope to spur the application of these standards to social-network data in a practical fashion. In future work we plan to study the application of out-link privacy to other social-network analysis tasks and provide studies of these approaches on real-world network data.

ACKNOWLEDGMENT

This work was supported by the Center for the Science of Information, an NSF Science and Technology Center.

REFERENCES

[1] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, 2008, pp. 111–125.

[2] E. Zheleva and L. Getoor, "Privacy in Social Networks: A Survey," in *Social Network Data Analytics*, Aggarwal, C. C., Ed., 2011, p. 277.

[3] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," *2009 30th IEEE Symposium on Security and Privacy*, vol. 0, no. c, pp. 173–187, 2009.

[4] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, ser. Lecture Notes in Computer Science, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Springer Berlin / Heidelberg, 2008, pp. 1–19.

[5] M. Hay, V. Rastogi, G. Miklau, and D. Suciu, "Boosting the accuracy of differentially private histograms through consistency," *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 1021–1032, Sep. 2010.

[6] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," *Data Mining, IEEE International Conference on*, pp. 169–178, 2009.

[7] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev, "Private analysis of graph structure," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, 2011.

[8] P. Marsden, "Network data and measurement," *Annual review of sociology*, pp. 435–463, 1990.

[9] D. J. Watts and S. H. Strogatz, "Collective dynamics of "small-world" networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[10] P. Holland and S. Leinhardt, "Local structure in social networks," *Sociological methodology*, vol. 7, no. 1, 1976.

[11] A. Marin and B. Wellman, "Social network analysis: An introduction," *Handbook of social network analysis*, vol. 22, no. January, 2010.

[12] M. Newman, "The structure and function of complex networks," *SIAM review*, pp. 167–256, 2003.

[13] A. Degenne and M. Forsé, *Introducing social networks*. SAGE Publications Ltd, 1999.

[14] D. J. Mir and R. N. Wright, "A differentially private graph estimator," in *Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*. IEEE Computer Society, 2009, pp. 122–129.

[15] D. Proserpio, S. Goldberg, and F. McSherry, "A workflow for differentially-private graph synthesis," 2012.

[16] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao, "Sharing graphs using differentially private graph models," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. New York, NY, USA: ACM, 2011, pp. 81–98.

[17] A. Gupta, A. Roth, and J. Ullman, "Iterative constructions and private data release," in *TCC*, 2012, pp. 339–356.

[18] J. J. P. III, T. L. Fond, S. Moreno, and J. Neville, "Fast generation of large scale social networks with clustering," *CoRR*, 2012.

[19] A. Machanavajjhala, A. Korolova, and A. D. Sarma, "Personalized social recommendations: accurate or private," *Proc. VLDB Endow.*, vol. 4, no. 7, pp. 440–450, Apr. 2011.