Zero-Knowledge Private Graph Summarization

Maryam Shoaran
University of Victoria, BC, Canada
Email: maryam@cs.uvic.ca

Alex Thomo
University of Victoria, BC, Canada
Email: thomo@cs.uvic.ca

Jens H. Weber-Jahnke
University of Victoria, BC, Canada
Email: jens@cs.uvic.ca

***Abstract*—Graphs have become increasingly popular for modeling data in a wide variety of applications, and graph summarization is a useful technique to analyze information from large graphs. Privacy preserving mechanisms are vital to protect the privacy of individuals or institutions when releasing aggregate numbers, such as those in graph summarization. We propose privacy-aware release of graph summarization using zero-knowledge privacy (ZKP), a recently proposed privacy framework that is more effective than differential privacy (DP) for graph and social network databases. We first define group-based graph summaries. Next, we present techniques to compute the parameters required to design ZKP methods for each type of aggregate data. Then, we present an approach to achieve ZKP for probabilistic graphs.**

I. Introduction

Nowadays, the graphs of many real world datasets are very large. For example, Facebook, the most well-known social network, contains data for over 900 million users and their relationships. Therefore, effective summarization methods need to be employed in order to make the analysis of such large graphs possible. We focus on graph summarization based on attribute groups. For instance, the nodes of a social graph can be grouped by attributes age and profession, and statistics about the number of cross-group edges can be recorded. Statistics can reveal interesting facts about a graph. For instance, they could show surprising strong connections between groups of people in different age and profession groups. As such, group-based graph summarization (GGS) is a ubiquitous operation in virtually all the graph/social network software products (cf. [4], [3], [1], [2], etc) The result of GGS is a smaller summary graph, where each node summarizes a group of nodes in the original graph.

The problem is that, as with other aggregations, the summary graphs are often released to other parties for further research purposes, and this brings up the matter of privacy. Aggregate data included in a summary graph can reveal sensitive and private information about the nodes (participants) of the underlying graph (network).

Privacy-preserving data release has become one of the most important problems today. ϵ-Differential Privacy [13], [11], [12] (DP for short) has been one of the leading privacy mechanisms in recent years. DP provides privacy for an individual of interest (IOI) by adding random noise to numerical outputs.

Some recent studies (cf. [14], [19]), however, have highlighted situations in which DP might not provide sufficient privacy protection. This is especially pronounced in social networks where different types of auxiliary information, including the structure of network or the groups the individuals belong in, are often readily available to the public (cf. [14]).

More specifically, whereas the goal of DP is to protect the participation of an individual (or relationship) in a dataset, in social networks we also need to protect the *evidence of participation* (cf. [19]). To see this we present the following example. Suppose there are two groups g_1 and g_2 and we want to publish the number of edges between them. Bob, a member of g_1, has an edge to Alice, a member of g_2. As a consequence of this connection, some friends of Bob introduce edges to Alice. What we want to protect is Bob's edge to Alice. DP works in this case by ensuring that for any true answer, c or $c-1$, the sanitized answer would be pretty much the same. However, this is not strong enough; the existence of Bob's edge changes the true answer not just by 1, but by a bigger number as it causes more edges to be created between the two groups.

Going beyond differential privacy, Gehrke, Lui, and Pass proposed "zero-knowledge privacy" (ZKP) in [14], which provides stronger privacy, especially for social graphs. The definition of ZKP is based on classes of aggregate functions. ZKP guarantees that an attacker cannot discover any personal information more than what can be inferred from some aggregate on a sample of a database with IOI removed. The sample complexity defines the level of privacy tolerance in ZKP. For instance, suppose in the Bob's example above the network size is 10000 and the sample size is $\sqrt{10000} = 100$. With such a sampling rate of 0.01 the evidence provided by say 10 more edges caused by Bob's edge will essentially be protected; with a high probability, none of these 10 edges will be in the sample.

In this paper, we use ZKP to provide individual privacy in graph summarization. We address connection measures for groups in social graphs and present ZKP mechanisms for private release of such aggregate outputs. To the best of our knowledge, we are the first to use ZKP for group-based graph summarizations, which are ubiquitous in analyzing social graphs.

As ZKP inherently depends on the precise characterization of sample complexity, we propose methods to compute the sample complexity of our aggregate functions. In order to achieve this, we present techniques to express the aggregate functions as averages of specially designed, synthetic attributes on the nodes of the graph. Then we derive precise prescriptions on how to construct ZKP mechanisms for the aggregate functions we consider.

More specifically, our contributions in this paper are as follows.

- We define group connection measures for graph summarization and consider different scenarios for zero-knowledge private release of summary graphs based on the type of personal information that is to be protected. We introduce synthetic attributes that simplify the construction and analysis of ZKP-mechanisms for graph summarization.

- We present detailed examples and numeric evaluations of our ZKP mechanisms in terms of the parameters involved. These evaluations are valid for any case and illustrate the trade offs involved when building ZKP mechanisms for graph summarization.
- We also present summarization measures for probabilistic graphs. This is especially important in social networks having edges of different influence captured by probabilities assigned on the edges.

II. Related Work

Graph summarization is a ubiquitous method for analyzing large graphs. Virtually all the graph/social network products (cf. [4], [3], [1], [2], etc) create summaries in the form of smaller graphs by grouping the nodes based on attributes.

The common goal of privacy preserving methods is to learn from data while protecting sensitive information of the individuals. k-anonymity for social graphs (cf. [22], [8], [9]) provides privacy by ensuring that combinations of identifying attributes appear at least k times in the dataset. The problem with k-anonymity and other related approaches, e.g. l-diversity [23], is that they assume the adversary has limited auxiliary knowledge. Narayanan and Shmatikov [25] present a de-anonymization algorithm and claim that k-anonymity can defeated by their method using auxiliary data accessible by the adversary.

Among a multitude of different techniques, differential privacy (DP) [6], [10], [13], [11] has become one of the leading methods to provide individual privacy. Various differentially private algorithms have since been developed for different domains, including social networks [16], [26]. However as already shown, DP can suffer in social networks where specific auxiliary information, such as graph structures and friendship data, is easily available to the adversary. Important works showing the shortcomings of DP are [19], [20].

Gehrke, Lui, and Pass in [14] present the notion of zero-knowledge privacy which is appealing for achieving privacy in social networks. Zero-knowledge privacy (ZKP) guarantees that what can be learned from a dataset including an individual is not more than what is learned from sampling-based aggregates computed on the dataset without that individual.

III. Graph Summarization

We denote a graph as $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges connecting the nodes. We consider $\mathcal{S} \subset 2^V$ to be a set of disjoint node groups of size r or more that a social network wants to release statistics about.

Definition 1: The $\mathcal{S}$*-graph* of G is $\mathcal{G}_{G,\mathcal{S}} = (\mathcal{S}, \mathcal{E}_\mathcal{S})$, where

$$\mathcal{E}_\mathcal{S} = \{(g', g'') \quad : \quad g', g'' \in \mathcal{S} \text{ and } \exists v' \in g' \text{ and } \exists v'' \in g'' \text{ such that } (v', v'') \in E\}.$$

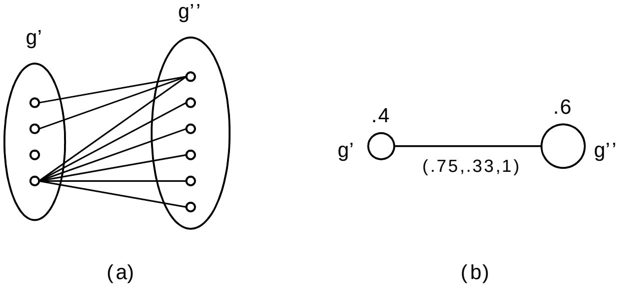

Fig. 1. A graph and its summarization.

This definition says that two groups g' and g'' in $\mathcal{S}$ are connected through an edge in $\mathcal{G}_{G,\mathcal{S}}$ if there exists at least one edge in G that connects a node in g' to a node in g''.

Definition 2: The $\mathcal{S}$*-graph summarization ($\mathcal{S}$-GS)* is a function

$$\begin{aligned}
w_1 &: \mathcal{S} \longrightarrow [0,1] \\
w_2 &: \mathcal{E}_\mathcal{S} \longrightarrow [0,1] \times [0,1] \times [0,1] \\
w_1(g) &= \frac{|g|}{|V|} \\
w_2(g', g'') &= (x, y, z), \text{ where} \\
x &= \frac{|\{v' \in g' : \exists v'' \in g'', \text{ s.t. } (v', v'') \in E\}|}{|g'|} \\
z &= \frac{|\{v'' \in g'' : \exists v' \in g', \text{ s.t. } (v', v'') \in E\}|}{|g''|} \\
y &= \frac{|\{(v', v'') : v' \in g', v'' \in g'', (v', v'') \in E\}|}{|g'| \cdot |g''|}.
\end{aligned}$$

Throughout the paper, we will refer to the elements of w_2 as $w_2(g', g'')[x]$, $w_2(g', g'')[y]$, and $w_2(g', g'')[z]$, or $w_2[x]$, $w_2[y]$, $w_2[z]$, whenever g' and g'' are clear from the context. We will also use $w_2[.]$ to refer to any of three elements x, y, or z.

Example 1: Fig.1 (a) shows a simple graph G, and $\mathcal{S}$ consisting of two groups g' and g''. Group g' has four nodes and group g'' has six nodes. There are several edges (eight of them) connecting members of g' to members of g''.

Fig.1 (b) shows how g' and g'' are represented by a node each in $\mathcal{G}_{G,\mathcal{S}}$. The nodes and the edge connecting them in $\mathcal{G}_{G,\mathcal{S}}$ are labeled by w_1 and w_2 measures, respectively, as described above. Specifically, we have $w_1(g') = \frac{4}{4+6} = .4$ and $w_1(g'') = \frac{6}{4+6} = .6$. Since three out of four nodes in g' and all the six nodes in g'' are connected with some nodes in the other group, we have $w_2(g', g'') = (\frac{3}{4}, \frac{8}{4\times6}, \frac{6}{6}) = (.75, .33, 1)$.

IV. Background on ϵ-Zero-Knowledge Privacy

Zero-Knowledge Privacy (ZKP) introduced by [14] is a privacy framework that is stronger than *Differential Privacy* (*DP*). ZKP is especially desirable in social networks where we need to protect not only the participation of a connection, but also easy to find evidence of the participation, as for example, the evidence given by other connections that were influenced by the connection.

ZKP is defined in relation with classes of sampling-based aggregate information. The class of sampling-based aggregation represents our tolerance for information release. For example, we can say that we are only comfortable to release the average age of a population computed on a $\sqrt{n}$ random sample. ZKP uses the notion of a simulator from zero-knowledge, and says that a simulator with the acceptable aggregate information can essentially compute whatever an adversary can compute by accessing the result of the mechanism ([14]). We describe ZKP in the following using a setting of graphs.

Let G be a graph. We denote by G_{-*} a graph obtained from G by removing a piece of information (for example an edge). G and G_{-*} are called *neighboring graphs*.

Let San be a mechanism that operates on a graph G (the complete database), and computes a *sanitized* answer to a query. The adversary's goal in a privacy scenario is to gain information about private matters of individuals (nodes) or connections (edges) in G using this released sanitized answer. Let $Adv(San(G), z)$ denote the output of the algorithm that an adversary employs to breach privacy. The adversary can interact with mechanism San and may have access to some auxiliary information z. The information in z is considered to be general, and easily accessible, e.g. information about the structure of the network (graph) or the groups that individuals belong in.

Let agg be a class of randomized algorithms that first select $k = k(n)$ random samples (nodes) without replacement from G_{-*}, and then compute some aggregate information. Such algorithms output an approximate answer to the query.

Let Sim, "the simulator," be an algorithm. We denote by $Sim(T(G_{-*}), z)$ the information that the simulator can compute given the aggregate information computed by a $T \in agg_k$. In plain language, imagine Sim to be a person who can be "extremely smart and capable (ESC)" and who has access to aggregates computed by the algorithms of class agg on the database where the sensitive information has been removed. Also, assume that the simulator also has access to background information z.

On the other hand, imagine the adversary to be a person who is also ESC, and has access to $San(G)$ as well as background information z. ZKP assures an individual that the participation in the network does not jeopardize her/his privacy. ZKP provides this guarantee by sanitizing the query answers such that the information that the adversary could extract from the output (sanitized answer) is computationally indistinguishable from the information that could be computed using sampling-based aggregates calculated on the network data that misses the individual of interests's sensitive information. That is, the adversary is not better off than some simulator even though he has access to the output of mechanism San computed on the whole database.

Definition 3: (Zero-Knowledge Privacy *[14]*) The mechanism San is **ϵ-zero-knowledge private with respect to** agg if there exists a $T \in agg$ such that for every adversary Adv, there exists a simulator Sim such that for every G, every $z \in \{0,1\}^*$, and every $W \subseteq \{0,1\}^*$, the following hold:

$$\begin{aligned} Pr[Adv(San(G), z) \in W] &\leq e^{\epsilon} \cdot Pr[Sim(T(G_{-*}), z) \in W] \\ Pr[Sim(T(G_{-*}), z) \in W] &\leq e^{\epsilon} \cdot Pr[Adv(San(G), z) \in W] \end{aligned}$$

where probabilities are taken over the randomness of San and Adv, and T and Sim.

By this definition, ZKP guarantees that any additional information that an adversary can obtain about an individual by having access to the output of the mechanism is virtually not more than what can be computed by a simulator using some sampling-based (approximate) aggregates even without access to the mechanism and the sensitive data.

Note that the selection of k – the number of random samples – in agg algorithms is very important and it should be chosen so that with high probability very few of the nodes connected with the node whose information has to be private will be chosen. We will often index agg by k as agg_k to stress the importance of k. To satisfy the ZKP definition, a mechanism should use $k = o(n)$, say $k = \sqrt{n}$ or $k = \sqrt[3]{n^2}$, where n, the number of nodes in the database, is sufficiently large (see [14]). DP is a special case of ZKP where $k = n$.

As it will be illustrated in the upcoming sections, the specifications of algorithm $T \in agg$, e.g. sample size, are only used to compute the level of privacy needed in the ZKP mechanism. We stress that the ZKP mechanism is the only algorithm applied on the data. The simulator is only an abstract notion.

Achieving ZKP. Let $f : \mathbf{G} \to \mathbb{R}^m$ be a function that produces a vector of length m from a graph database. For example, given $\mathcal{G}_{G,\mathcal{S}}$, f produces the results of the $\mathcal{S}$-GS functions, i.e. w on edges.

We consider the L_1-Sensitivity to be defined as follows.

Definition 4: (L_1-Sensitivity) For $f : \mathbf{G} \to \mathbb{R}^m$, the L_1-sensitivity of f is

$$\Delta(f) = \max_{G', G''} ||f(G') - f(G'')||_1$$

for all neighboring graphs G' and G''.

Another essential definition is that of "sample complexity".

Definition 5: (Sample Complexity *[14]*) A function $f : Dom \to \mathbb{R}^m$ is said to have (δ, β)**-sample complexity** with respect to agg if there exists an algorithm $T \in agg$ such that for every $D \in Dom$ we have

$$Pr[||T(D) - f(D)||_1 \leq \delta] \geq 1 - \beta.$$

T is said to be a (δ, β)-sampler for f with respect to agg.

This definition bounds the probability of error between the randomized computation (approximation) of function f and the expected output of f. Basically, functions with low sample complexity (smaller δ and β) can be computed more accurately using random samples from the input data.

When the released information, as typical, is real numbers, the ZKP mechanism San achieves the privacy by adding noise to each of the numbers independently.

Let $Lap(\lambda)$ be the zero-mean Laplace distribution with scale λ, and variance $2\lambda^2$. The scale of Laplace noise in ZKP is properly calibrated to the sample complexity of the function that is to be privately computed. The following proposition expresses the relationship between the sample complexity of a function and the level of zero knowledge privacy achieved by adding Laplace noise to the outputs of the function.

Proposition 1: ([14]) Suppose $f : \mathbf{G} \to [a, b]^m$ has (δ, β)-sample complexity with respect to agg. Then, mechanism

$$San(G) = f(G) + (X_1, \ldots, X_m),$$

where $G \in \mathbf{G}$, and $X_j \backsim Lap(\lambda)$ for $j = 1, \cdots, m$ independently, is

$$\ln\left((1-\beta)e^{\frac{\Delta(f)+\delta}{\lambda}} + \beta e^{\frac{(b-a)m}{\lambda}}\right)$$

–ZKP with respect to agg.

V. ZKP Mechanism for Graph Summarization

In this section we design a ZKP mechanism to release a graph summarization. Let $\mathcal{G}_{G,\mathcal{S}} = (\mathcal{S}, \mathcal{E}_\mathcal{S})$ be the $\mathcal{S}$-graph for a graph G. Let f be a function that takes a graph $\mathcal{G}_{G,\mathcal{S}}$ as input and produces $c = |\mathcal{S}| + 3 \cdot |\mathcal{E}_\mathcal{S}|$ numbers, or differently said, a c-dimensional vector corresponding to w_1 and w_2 aggregates for the groups and the connecting edges.

Let $f = [f_1, \ldots, f_t]$ be the vector (or subvector) that is to be privately released. We apply a separate San_i (ZKP) mechanism, for $i \in [1, t]$, to each of the elements of f. Let us assume that each San_i provides ϵ_i-ZKP for f_i with respect to agg_{k_i}, where $k_i = k(n)/t$ and $n = |V|$. Then, based on the following proposition, f will be $(\sum_{i=1}^{t} \epsilon_i)$-ZKP with respect to $agg_{k(n)}$, where $k(n) = \sum_{i=1}^{t} k_i$.

Proposition 2: (Sequential Composition [14]) Suppose San_i, for $i \in [1, n]$, is an ϵ_i-ZKP mechanism with respect to agg_{k_i}. Then, the mechanism resulting from composing[1] San_i's is $(\sum_{i=1}^{n} \epsilon_i)$-ZKP with respect to $agg_{(\sum k_i)}$.

In this paper, we consider edge (connection) privacy. We note that node privacy will not be considered in this work, since, as it is widely considered (cf. [16], [19]), it results in too noisy output with practically no utility.

A. Edge (Connection) Privacy

Consider $\mathcal{G}_{G,\mathcal{S}}$ and $\mathcal{G}_{G_{-e},\mathcal{S}}$, where $\mathcal{G}_{G_{-e},\mathcal{S}}$ is a neighboring graph of G obtained from G by removing edge e. In the edge privacy scenario, the total number of nodes (groups) and the size of each group are identical in $\mathcal{G}_{G,\mathcal{S}}$ and $\mathcal{G}_{G_{-e},\mathcal{S}}$. Therefore, the sensitivity of any w_1 function, including the ones in f, is zero, that is, $\Delta(w_1) = 0$.

On the other hand, removing an edge in G can change by at most 1 the numerator of each element x, y, and z in w_2 measures of $\mathcal{G}_{G_{-e},\mathcal{S}}$. Note that this change affects only one w_2 measure in the whole graph $\mathcal{G}_{G_{-e},\mathcal{S}}$. Therefore, the sensitivities of the elements of any w_2 function, including the ones in f, are

$$\Delta(w_2[x]) = \tfrac{1}{r} \qquad \Delta(w_2[y]) = \tfrac{1}{r^2} \qquad \Delta(w_2[z]) = \tfrac{1}{r}$$

where r is the minimum group size in $\mathcal{S}$.

In the following sections, the ZKP mechanisms are separately designed for w_1 and w_2 functions in f.

1) ZKP Mechanism for w_1: Suppose $w_1(g)$ is an element of f, where g is a group in $\mathcal{G}_{G,\mathcal{S}}$. Let $San = w_1(g) + Lap(\lambda)$ be a ZKP mechanism which adds random noise selected from $Lap(\lambda)$ distribution to the output of $w_1(g)$ in order to achieve ZKP. Our goal here is to come up with the right λ to achieve a predefined level of ZKP.

Based on the definition of ZKP, one should first know the sample complexity of the w_1 function. For this, without change in semantics, we will express w_1 so that it computes an average rather than a fraction of two counts. Then, using the *Hoeffding* inequality (cf. [24]) we compute the sample complexity of w_1.

Expressing w_1. We assume that in addition to the regular node attributes (if any), we have $|\mathcal{S}|$ new boolean attributes, one for each possible group. We denote these new attributes by uppercase B's indexed by the group id. A node v in graph G will have $B_g(v) = 1$ if v belongs to group g, and $B_g(v) = 0$, otherwise. We have that,

Proposition 3:

$$w_1(g) = \frac{\sum_{v \in V} B_g(v)}{|V|}.$$

Therefore, $w_1(g)$ can be viewed as the average value of attribute B_g over all nodes in G.

ZKP Mechanism. Let $G = (V, E)$ be a graph enriched with boolean attributes as explained above. We would like to determine the value of $\lambda > 0$ for $Lap(\lambda)$ distribution which is to be used to add random noise to a $w_1(g)$ measure included in f. For this, first we compute the sample complexity of w_1 to be able to use Proposition 1 and establish an appropriate value for λ.

Let T be a randomized algorithm in agg_k, the class of randomized algorithms that operates on an input graph G. To randomly sample a graph G, algorithm T uniformly selects $k = k(n)/t$ random nodes from V, reads their attributes, and retrieves all the edges[2] incident to these k sample nodes.[3] From the sampled nodes and their incident edges with other sampled nodes we consider $\mathcal{G}_{G',\mathcal{S}'} = (\mathcal{S}', \mathcal{E}'_\mathcal{S})$. Then, T approximates the value of $w_1(g)$ using $\mathcal{G}_{G',\mathcal{S}'}$. Since we have expressed $w_1(g)$ for a group g as an average, based on the Hoeffding inequality we have

$$Pr[|T(g) - w_1(g)| \leq \delta] \geq 1 - 2e^{-2k\delta^2}.$$

From this and Definition 5, we have that w_1 has $\left(\delta, 2e^{-2k\delta^2}\right)$-sample complexity with respect to agg_k.

Now we make the following substitutions in the formula of Proposition 1: $\beta = 2e^{-2k\delta^2}$, $\Delta(w_1(g)) = 0$, $b - a = 1$, and $m = 1$ and obtain that mechanism San is

$$\ln\left(e^{\frac{\delta}{\lambda}} + 2e^{\frac{1}{\lambda} - 2k\delta^2}\right) \text{ -ZKP}$$

with respect to agg_k.

[1] A set of computations that are separately applied on *one* database and each provides ZKP in isolation, also provides ZKP for the set.

[2] Clearly, only non-dangling incident edges, whose both end nodes have been sampled, will be retrieved.

[3] For other possible methods of graph sampling see for example [14].

Similarly to DP, one can set λ, the Laplace noise scale, to be proportional to "the error" as measured by the sum of the sensitivity and sampling error, and inversely proportional to the ZKP privacy level

$$\lambda = \frac{\Delta(w_1)+\delta}{\epsilon} = \frac{1}{\epsilon}\cdot\frac{1}{\sqrt[3]{k}}.$$

Regarding δ, we can consider for instance a sample size $k = \sqrt[3]{n^2}$, and have $\delta = \frac{1}{\sqrt[3]{k}}$.

From all the above, the privacy level obtained will be

$$\begin{aligned}\ln\left(e^{\frac{\delta}{\lambda}}+2e^{\frac{1}{\lambda}-2k\delta^2}\right) &= \ln\left(e^{\epsilon}+2e^{\epsilon\sqrt[3]{k}-2\sqrt[3]{k}}\right)\\ &\leq \ln\left(e^{\epsilon}+2e^{-\sqrt[3]{k}}\right)\\ &\leq \epsilon+2e^{-\sqrt[3]{k}}.\end{aligned}$$

Thus, we have that by adding $Lap\left(\frac{1}{\epsilon\cdot\sqrt[3]{k}}\right)$ noise, mechanism San will be $\left(\epsilon+2e^{-\sqrt[3]{k}}\right)$-ZKP with respect to agg_k. Of course, the privacy achieved is in fact better than this because of the above inequalities. We address finding of the exact λ given a ZKP privacy level and sample complexity in Section VII.

Example 2: Let graph G be a social graph with one hundred million participants/nodes ($|V| = n = 100,000,000$), and g', g'' be two groups. Suppose the requested output vector is

$$f = \langle w_1(g'), w_2(g',g'')[x], w_2(g',g'')[y], w_2(g',g'')[z], w_1(g'')\rangle.$$

and suppose that the minimum group size in $\mathcal{S}$ is $r = 5000$. Assume we would like to have for f a ZKP mechanism expressed with respect to an acceptable agg_k, where

$$k(n) = \sqrt[3]{100,000,000^2} = 215,443.$$

To privately release the first output in f, a randomized algorithm T can uniformly select

$$k_1 = 215,443/5 = 43,089.$$

nodes and approximate the value of $w_1(g')$ using these samples. Let (δ_1,β_1) be the sample complexity of $w_1(g')$ where

$$\delta_1 = \frac{1}{\sqrt[3]{k_1}} = \frac{1}{\sqrt[3]{43,089}} = 0.0285$$

$$\beta_1 = 2e^{-2k_1\delta_1^2} = 2e^{-2*43089*(0.0285)^2} = 7.97*10^{-31}.$$

The sensitivity of f is

$$\Delta(f) = \frac{1}{r}+\frac{1}{r^2}+\frac{1}{r} = 0.0004.$$

Now, if we would like to use a mechanism which is 0.1-ZKP, we add random noise selected from a Laplace distribution with scale

$$\lambda_1 = \frac{\Delta(f)+\delta_1}{\epsilon} = \frac{0.0004+0.0285}{0.1} = 0.289$$

to the actual value of $w_1(g')$. With this noise scale, the ZKP privacy level of the mecahnism is precisely

$$\epsilon_1 \leq \left(\epsilon+2e^{-\sqrt[3]{k_1}}\right) = \left(0.1+2e^{-35.06}\right) \approx 0.1$$

with respect to agg_k.

2) ZKP Mechanism for w_2: Suppose the function $w_2(g,g')[.]$ is an element of f, where g and g' are groups in $\mathcal{G}_{G,\mathcal{S}}$. Let $San = w_2(g,g')[.] + Lap(\lambda)$ be a ZKP mechanism that adds random noise selected from $Lap(\lambda)$ distribution to $w_2(g,g')[.]$. To come up with the right λ first we compute the sample complexity of the w_2 function.

Expressing $w_2[x]$ and $w_2[z]$. To express the x or z elements of the w_2 function, we introduce $|\mathcal{S}|$ new boolean node attributes, each corresponding to a group. We denote these new attributes by B' indexed by the group id. A node v will have $B'_g(v) = 1$ if v has an edge with some node in group g, and $B'_g(v) = 0$, otherwise. Now for each pair of groups g and g' we can show the following proposition.

Proposition 4:

$$\begin{aligned}w_2(g,g')[x] &= \frac{\sum_{v\in g} B'_{g'}(v)}{|g|}\\ w_2(g,g')[z] &= \frac{\sum_{v\in g'} B'_{g}(v)}{|g'|}\end{aligned}$$

Hence, the x (or z) elements of $w_2(g,g')$ can be viewed as the average value of attribute $B'_{g'}$ (or B'_g) over the subset of nodes in G that are in g' (or g).

Expressing $w_2[y]$. To express y in w_2, we introduce $|\mathcal{S}|$ new node attributes, each corresponding to a group. We denote these new attributes by B'' indexed by the group id. Each attribute B''_g is a boolean vector of dimension $|g|$, where each dimension corresponds to a node in g. A node v will have $B''_g(v)[u] = 1$, where $u \in g$, if (v,u) is an edge in graph G, and $B''_g(v)[u] = 0$, otherwise. For each pair of groups g and g' we can show that

Proposition 5:

$$\begin{aligned}w_2(g,g')[y] &= \frac{\sum_{v\in g, u\in g'} B''_{g'}(v)[u]}{|g|\cdot|g'|}\\ &= \frac{\sum_{v\in g', u\in g} B''_{g}(v)[u]}{|g|\cdot|g'|}\end{aligned}$$

Therefore, the y measure in $w_2(g,g')$ can be viewed as the average of $B''_{g'}(v)[u]$'s or $B''_g(v)[u]$'s.

ZKP Mechanism. Let $G = (V,E)$ be a graph enriched with boolean attributes as explained above. We would like to determine the value of $\lambda > 0$ for the $Lap(\lambda)$ distribution which will add random noise to $w_2(g,g')[.]$.

Let T be a randomized algorithm in agg_k. Algorithm T randomly samples graph G by uniformly selecting $k = k(n)/t$ random nodes from V and retrieving all the incident edges. With this sampling, the nodes in the groups of $\mathcal{G}_{G,\mathcal{S}}$ and the edges between them are randomly sampled as well. We call this sampled $\mathcal{S}$-graph $\mathcal{G}'_{G,\mathcal{S}} = (\mathcal{S}',\mathcal{E}'_{\mathcal{S}})$. Let us assume that we have a sample of each group and edges between groups and the size of a sample group g is k_g. Then, algorithm T approximates w_2 using the data from group samples. For the sample complexity of the elements of w_2, since we expressed them as averages, we can use the Hoeffding inequality as follows.

$$\begin{aligned} Pr[|T(g,g')[x] - w_2(g,g')[x]| \leq \delta] &\geq 1 - 2e^{-2k_g\delta^2} \\ Pr[|T(g,g')[z] - w_2(g,g')[z]| \leq \delta] &\geq 1 - 2e^{-2k_{g'}\delta^2} \\ Pr[|T(g,g')[y] - w_2(g,g')[y]| \leq \delta] &\geq 1 - 2e^{-2(k_g \times k_{g'})\delta^2}. \end{aligned}$$

Let us focus first on $w_2[x]$ ($w_2[z]$ is similar). Now we make the following substitutions in the formula of Proposition 1: $\beta = 2e^{-2k_g\delta^2}$, $\Delta(w_2(g,g')[x]) = 1/r$, $b - a = 1$, and $m = 1$. From this, we have that mechanism San is

$$\ln\left(e^{\frac{1/r+\delta}{\lambda}} + 2e^{\frac{1}{\lambda} - 2k_g\delta^2}\right) \text{ -ZKP}$$

with respect to agg_k.

Again, one can set λ, the Laplace noise scale, to be proportional to "the error" as measured by the sum of the sensitivity and sampling error, and inversely proportional to ϵ

$$\lambda = \frac{\Delta(w_2)[x] + \delta}{\epsilon} = \frac{1}{\epsilon}\left(\frac{1}{r} + \frac{1}{\sqrt[3]{k_g}}\right)$$

Regarding δ, we can consider for instance a sample size $k = \sqrt[3]{n^2}$, and have $\delta = \frac{1}{\sqrt[3]{k_g}}$.

From all the above, the privacy level obtained will be

$$\begin{aligned} \ln\left(e^{\frac{1/r+\delta}{\lambda}} + 2e^{\frac{1}{\lambda} - 2k_g\delta^2}\right) &= \ln\left(e^{\epsilon} + 2e^{\frac{\epsilon}{1/r+1/\sqrt[3]{k_g}} - 2\sqrt[3]{k_g}}\right) \\ &\leq \ln\left(e^{\epsilon} + 2e^{-\sqrt[3]{k_g}}\right) \\ &\leq \epsilon + 2e^{-\sqrt[3]{k_g}}. \end{aligned}$$

Thus, we have that by adding noise randomly selected from the $Lap\left(\frac{1}{\epsilon}\left(\frac{1}{r} + \frac{1}{\sqrt[3]{k_g}}\right)\right)$ distribution to $w_2[x]$, San will be $\left(\epsilon + 2e^{-\sqrt[3]{k_g}}\right)$-ZKP with respect to agg_k.

By substituting for the proper sensitivity and sample complexity, similar computations can be carried out for a San mechanism for $w_2[y]$.

Example 3: Let us consider Example 2 again with the output vector

$$f = \langle w_1(g'), w_2(g',g'')[x], w_2(g',g'')[y], w_2(g',g'')[z], w_1(g'')\rangle.$$

To privately release the second output in f, a randomized algorithm T can again uniformly select

$$k_2 = k(n)/5 = \sqrt[3]{100{,}000{,}000^2}/5 = 43{,}089$$

nodes and approximate the value of $w_2(g',g'')[x]$. The actual value of function $w_2(g',g'')[x]$ is computed on G. Suppose that the minimum group size in $\mathcal{S}$ is $r = 5000$ and the size of the sample group corresponding to g' in $\mathcal{G}'_{G,S}$ is $k_{g'} = 50000$. Let (δ_2, β_2) be the sample complexity of $w_2(g',g'')[x]$ where

$$\delta_2 = \frac{1}{\sqrt[3]{k_{g'}}} = \frac{1}{\sqrt[3]{50000}} = 0.0271.$$

$$\beta_2 = 2e^{-2k_{g'}\delta_2^2} = 2e^{-2*50000*(0.0271)^2} = 2.54 * 10^{-32}.$$

The sensitivity of f is

$$\Delta(f) = \frac{1}{r} + \frac{1}{r^2} + \frac{1}{r} = 0.0004.$$

Now, if we would like to use a mechanism which is 0.1-ZKP, we can add random noise selected from a Laplace distribution with scale

$$\lambda_2 = \frac{\Delta(f) + \delta_2}{\epsilon} = \frac{0.0004 + 0.0271}{0.1} = 0.275$$

to the actual value of $w_2(g',g'')[x]$.

With this noise scale, the ZKP privacy level of the mecahnism is precisely

$$\epsilon_2 \leq \left(\epsilon + 2e^{-\sqrt[3]{k_{g'}}}\right) = \left(0.1 + 2e^{-36.84}\right) \approx 0.1$$

with respect to agg_k.

VI. Evaluation

We focus on a single output $w_1(g)$ to evaluate our approach (the evaluation based on w_2 is similar). In our methods, the amount of noise added to the output is independent of the database, and it only depends on the aggregates we compute and their sensitivities. Therefore, the following analysis is valid for any database.

A. Parameters Affecting Noise Scale

Considering the formula of noise scale $\lambda = \frac{\Delta(f)+\delta}{\epsilon}$, the sampling error δ is an important factor specifying λ. The error in turn has reverse connection with the sample size and the size of the database graph. Recall that throughout the paper we considered the error to be $\delta = \frac{1}{\sqrt[3]{k}}$, where k is the sample size with values for example $k = \sqrt[3]{n^2}$.

Fig. 2 illustrates the relationship between the noise scale λ and the sample size k and the database size n. In this figure we assumed that the output vector f has five elements and the ZKP-level ϵ is 0.1. Each curve in the figure corresponds to a sample size, namely, $k = \sqrt[3]{n^2}$ and $k = \sqrt[4]{n^3}$. The figure shows that as the graph size decreases from one billion to one million the noise scale increases non-linearly to the amounts that are not practical in our setting. Therefore, we conclude that ZKP mechanisms are more practical in very big databases with sufficiently large sample size.

B. The Noise

The analysis in this section provides a better understanding of the amount of noise which is added to the output. We consider w_1 function.

We first compute the cumulative distribution function of Laplace distribution in an interval $[-z, z]$ as follows,

$$Pr(-z \leq x \leq z) = \int_{-z}^{z} \tfrac{1}{2\lambda} e^{\frac{-|x|}{\lambda}} dx = 1 - e^{\frac{-z}{\lambda}}.$$

Therefore, $Pr(|x| \geq z) = e^{\frac{-z}{\lambda}}$.

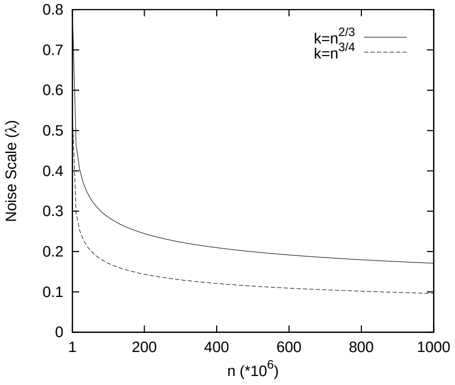

Fig. 2. Relationship between noise scale and database size.

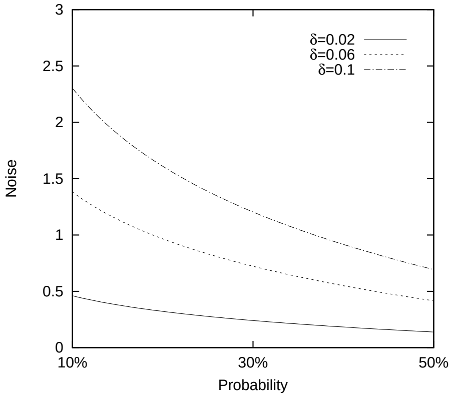

Fig. 3. Probability vs noise.

Let $pr = Pr(|x| \geq z)$. Value z for a specified cumulative probability pr can be calculated using the above equation as

$$z = -\lambda \cdot \ln(pr) = -\frac{\delta}{\epsilon} \cdot \ln(pr).$$

Figure 3 illustrates the minimum absolute noise z as a function of cumulative probability pr for three different values of δ when $\epsilon = 0.1$. Each point (pr, z) on the curve for a given δ means that

> pr percent of the time the random noise has an absolute value of at least z.

For example, for $\delta = 0.02$, we have that 50% of the time the absolute value of noise is at least 0.14, and 30% of the time it is at least 0.24. These values of δ are practical as our outputs are fractions.

VII. From Privacy Level to Noise Scale

In this section we address the problem of computing the noise scale based on the required privacy. For a given privacy level ϵ, the right value for λ can be computed using Proposition 1, and the sample complexity of the function. For this, we need to solve the following equation with respect to λ.

$$\ln\left((1-\beta) \cdot e^{\frac{\Delta(f)+\delta}{\lambda}} + \beta \cdot e^{\frac{(b-a)m}{\lambda}}\right) \quad = \quad \epsilon.$$

The sample complexity (δ, β) of the function and the sensitivity $\Delta(f)$ are computed as described in Section V. Suppose $b - a$ and m are both equal to 1 (as in Section V). Thus, by assigning δ a value (which depends on k), λ is the only variable in this equation. By setting $x = e^{\frac{1}{\lambda}}$, we have the polynomial equation

$$(1-\beta)x^{\Delta(f)+\delta} + \beta x - e^{\epsilon} = 0$$

which can be solved for x using various methods (cf. [18], [15]), and finally, we have $\lambda = \frac{1}{\ln x}$.

Example 4: Let us consider Example 2 again with the output vector

$$f = \langle w_1(g'), w_2(g', g'')[x], w_2(g', g'')[y], w_2(g', g'')[z], w_1(g'')\rangle.$$

Suppose that we would like to design a (0.1)-ZKP mechanism for $w_1(g')$.

To compute the corresponding noise scale λ_1, we use the above polynomial equation. We assume that the minimum group size in $\mathcal{S}$ is $r = 5000$. The sensitivity is $\Delta(f) = \frac{1}{r} + \frac{1}{r^2} + \frac{1}{r} = 0.0004$, and we consider $k_1 = k(n)/5 = \sqrt[3]{100,000,000^2/5} = 43,089$ (as in Example 2), i.e. $\delta = \frac{1}{\sqrt[3]{k_1}} = \frac{1}{\sqrt[3]{43089}} = 0.0285$. We have that $w_1(g')$ has a sample complexity of

$$\begin{aligned} (\delta, \beta) &= (\delta, 2e^{-2k_1\delta^2}) \\ &= (0.0285, 7.08 * 10^{-31}). \end{aligned}$$

Now if we plug all the values in the equation

$$(1-\beta)x^{\Delta(f)+\delta} + \beta x - e^{\epsilon} \quad = \quad 0$$

we have

$$\begin{aligned} &(1 - 7.08 * 10^{-31})x^{0.0004+0.0285} \\ &\qquad + (7.08^{-31})x - e^{0.1} = 0 \end{aligned}$$

which has root $x = 31.731745$. This results in a noise scale λ_1 that is very close to (albeit slightly lower than) what we computed in Example 2. Therefore, setting the noise scale to be proportional to $\Delta(f) + \delta$ and inversely proportional to ϵ is a good enough approximation for achieving $\epsilon - ZKP$.

VIII. Private Probabilistic A-GS

In this section we consider graphs with probabilistic edges. Such graphs are very common in modeling influences in social networks (cf. [17], [7], [21], [5]).

A. Probabilistic Graphs

We will denote a *probabilistic* graph by $G = (V, E)$, where V is the set of nodes, $E \subseteq V \times V$ is the set of edges, and additionally, assigned to each edge $e \in E$, there is an existence probability $p(e) \in [0, 1]$. A probabilistic graph defines a probability distribution over a set of deterministic (regular) graphs called *possible instances* (PIs). Let $\mathcal{PI}(G)$ (or simply $\mathcal{PI}$ when G is clear from the context) be the set of all PIs of a probabilistic graph G and $PI_i(G)$ (or simply PI_i) denote one single PI. The existence probability of each PI is computed as

$$p(PI) = \prod_{e \in E(PI)} p(e) \cdot \prod_{e \notin E(PI)} (1 - p(e)). \quad (1)$$

B. Probabilistic Graph Summarization

We define the summarization of a probabilistic graph G in a similar way as for deterministic graphs. We have a set of disjoint groups of nodes, and any two groups g and g' are connected in the summary graph if at least one edge connects a node from g to some node in g'. We denote the probabilistic summary graph corresponding to G as $\mathcal{S}$-GS.

Computing the w_1 measure does not change in the probabilistic case as it is based on the existence of nodes and their attribute values which none is probabilistic. However, due to probabilistic edges the numerators of x, y, and z of the w_2 measure are computed differently. That is, instead of computing the exact value, their *expected values* over the set of possible instances will need to be computed.

For this, let X, Y, and Z be random variables representing the x, y, and z measures, respectively. To compute $E[X]$ or $E[Z]$ we have the following theorem.

Theorem 1: Let g and g' be two groups in a probabilistic summary graph, and let $E_{v_j} = \{e_1, \ldots, e_{n_j}\}$ be the set of edges connecting a node $v_j \in g$ to the nodes of g'. We have that

$$E[X(g,g')] = E[X] = \frac{\sum_{v_j \in g} \left(1 - \prod_{e \in E_{v_j}} (1 - p(e))\right)}{|g|}.$$

For $E[Y]$, it can be verified that,

Theorem 2: Let V_g be the set of nodes in a group g and $E_{gg'} = V_g \times V_{g'}$ be the set of all possible edges between two groups g and g' in a probabilistic summary graph. We have that

$$E[Y(g,g')] = E[Y] = \frac{\sum_{e_i \in E_{gg'}} p(e_i)}{|g| \cdot |g'|}.$$

C. Zero-Knowledge Private Probabilistic A-GS

We focus on edge (connection) privacy in this section. Let $\mathcal{G}_{G,\mathcal{S}} = (\mathcal{S}, \mathcal{E}_\mathcal{S})$ be the probabilistic summary graph corresponding to a graph G. Let f be a subvector that is to be privately released.

As stated before, the w_1 elements in f are computed and privatized as illustrated in Section V-A1. For the elements of the $E[w_2]$ functions in f, we need to view them as averages to be able to use the Hoeffding inequality in the process of privatization. We do this by defining new synthetic attributes.

Expressing $E[Y]$. For each node $v \in V$, we assume to have $|\mathcal{S}|$ new attributes called P'', each indexed by a group id. Each attribute P''_g is a vector of dimension $|g|$, where each dimension corresponds to a node in g. For a node v we have $P''_g(v)[u] = p(e_{vu})$, where u is a node in g, and $p(e_{vu})$ is the probability of the edge between v and u. Clearly, $P''_g(v)[u] = 0$ if there is no edge between v and u in G.

For each pair of groups g and g' we have the following proposition.

Proposition 6:

$$\begin{aligned} E[Y(g,g')] &= \frac{\sum_{v \in g, u \in g'} P''_{g'}(v)[u]}{|g| \cdot |g'|} \\ &= \frac{\sum_{v \in g', u \in g} P''_{g}(v)[u]}{|g| \cdot |g'|} \end{aligned}$$

Note that, with this expression, $E[Y]$ is the average of the elements of attribute $P''_{g'}$ over the nodes of g, or vice versa.

Expressing $E[X]$ or $E[Z]$. To be able to view $E[X]$ or $E[z]$ functions in f as averages, for each node $v \in V$, we consider $\mathcal{S}$ new synthetic attributes called P', each indexed by a group id. For each attribute P'_g, we compute the attribute value as

$$P'_g(v) = 1 - \prod_{u \in g} (1 - P''_g(v)[u]) = 1 - \prod_{u \in g} (1 - p(e_{vu})).$$

Now for each pair of groups g and g' we have the following proposition.

Proposition 7:

$$\begin{aligned} E[X(g,g')] &= \frac{\sum_{v \in g} P'_{g'}(v)}{|g|} \\ E[Z(g,g')] &= \frac{\sum_{v \in g'} P'_{g}(v)}{|g'|} \end{aligned}$$

Clearly, $E[X(g,g')]$ $(E[Z(g,g')])$ is now the average of $P'_{g'}$ (P'_g) attribute over the nodes of g (g').

ZKP Mechanism. Let $G = (V, E)$ be a probabilistic graph augmented with synthetic attributes P's and P''s. To compute the sample complexity, a randomized algorithm T, in agg_k, samples graph G by uniformly selecting $k = k(n)/t$ random nodes from V and all their incident edges. Then, T approximates $E[w_2[.]]$ using the data from sample groups. Since we redefined the elements of $E[w_2[.]]$ as averages, we have the following inequalities for their sample complexities using the Hoeffding inequality.

$$\begin{aligned} Pr[|T(g,g')[x] - E[w_2(g,g')[X]]| \leq \delta] &\geq 1 - 2e^{-2k_g\delta^2} \\ Pr[|T(g,g')[z] - E[w_2(g,g')[Z]]| \leq \delta] &\geq 1 - 2e^{-2k_{g'}\delta^2} \\ Pr[|T(g,g')[y] - E[w_2(g,g')[Y]]| \leq \delta] &\geq 1 - 2e^{-2(k_g \times k_{g'})\delta^2} \end{aligned}$$

It can be verified that the sensitivities of $E[w_2[.]]$ functions are similar to the regular case. Thus, by plugging the above parameters in Proposition 1, we have the following for the San mechanism of $E[X]$, where r is the minimum group size in $\mathcal{S}$, and k_g is the size of a sample group g.

Proposition 8: By adding noise randomly selected from the $Lap\left(\frac{1}{\epsilon}\left(\frac{1}{r} + \frac{1}{\sqrt[3]{k_g}}\right)\right)$ distribution to the output of $E[X]$, San will be $\left(\epsilon + 2e^{-\sqrt[3]{k_g}}\right)$-ZKP with respect to agg_k.

A similar San mechanism can be proposed for $E[Y]$ by substituting for the sensitivity and sample complexity.

IX. Conclusions

We addressed zero-knowledge privacy for graph summarization. We focused on group connection measures that are supported by virtually all the social-graph software products. Our techniques are crucial to be applied on summary graphs before public release of the information. To the best of our knowledge, this is the first work to use the ZKP framework for graph summarization. We focused on ZKP mechanisms for edge privacy and introduced methods to compute the ZKP parameters. Furthemore, we presented an approach to achieve ZKP for the release of graph-summarization for probabilistic data. The upshot is that ZKP is quite useful for protecting not only the participation of a connection, but also the evidence of its participation. However, from a utility point of view, ZKP can only be applied meaningfully on big social graphs.

References

[1] Gephi. http://gephi.org/.

[2] Netdriller. http://pages.cpsc.ucalgary.ca/ nkoochak/NetDriller/.

[3] Nodexl. http://nodexl.codeplex.com/.

[4] Pajek. http://vlado.fmf.uni-lj.si/pub/networks/pajek/default.htm.

[5] S. Bhagat, A. Goyal, and L. V. S. Lakshmanan. Maximizing product adoption in social networks. In *WSDM*, pages 603–612, 2012.

[6] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In *PODS*, pages 128–138, 2005.

[7] C. Budak, D. Agrawal, and A. E. Abbadi. Limiting the spread of misinformation in social networks. In *WWW*, pages 665–674, 2011.

[8] S. Chester, B. M. Kapron, G. Ramesh, G. Srivastava, A. Thomo, and S. Venkatesh. k-anonymization of social networks by vertex addition. In *ADBIS*, pages 107–116, 2011.

[9] S. Chester, B. M. Kapron, G. Ramesh, G. Srivastava, A. Thomo, and S. Venkatesh. Why waldo befriended the dummy? k-anonymization of social networks with pseudo-nodes. *SNAM*, 2012.

[10] C. Dwork. Differential privacy. In *ICALP (2)*, pages 1–12, 2006.

[11] C. Dwork. Differential privacy: A survey of results. In *TAMC*, pages 1–19, 2008.

[12] C. Dwork. Differential privacy in new settings. In *SODA*, pages 174–183, 2010.

[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[14] J. Gehrke, E. Lui, and R. Pass. Towards privacy for social networks: A zero-knowledge based definition of privacy. In *TCC*, pages 432–449, 2011.

[15] S. Goedecker. Remark on algorithms to find roots of polynomials. 15(5):1059–1063, Sept. 1994.

[16] M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate estimation of the degree distribution of private networks. In *ICDM*, pages 169–178, 2009.

[17] J. J. P. III and J. Neville. Methods to determine node centrality and clustering in graphs with uncertain structure. In *ICWSM*, 2011.

[18] M. A. Jenkins. Algorithm 493: Zeros of a real polynomial [c2]. *ACM Trans. Math. Softw.*, 1(2):178–189, June 1975.

[19] D. Kifer and A. Machanavajjhala. No free lunch in data privacy. In *SIGMOD Conference*, pages 193–204, 2011.

[20] D. Kifer and A. Machanavajjhala. A rigorous and customizable framework for privacy. In *PODS*, pages 77–88, 2012.

[21] G. Kollios, M. Potamias, and E. Terzi. Clustering large probabilistic graphs. *IEEE Trans. Knowl. Data Eng.*, 2012.

[22] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *SIGMOD Conference*, pages 93–106, 2008.

[23] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. *L*-diversity: Privacy beyond *k*-anonymity. *TKDD*, 1(1), 2007.

[24] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.

[25] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, pages 173–187, 2009.

[26] V. Rastogi, M. Hay, G. Miklau, and D. Suciu. Relationship privacy: output perturbation for queries with joins. In *PODS*, pages 107–116, 2009.