Identity Obfuscation in Graphs Through the Information Theoretic Lens

Francesco Bonchi #1, Aristides Gionis #2, Tamir Tassa *3

# Yahoo! Research, Barcelona, Spain
1 bonchi@yahoo-inc.com
2 gionis@yahoo-inc.com

* Division of Computer Science, The Open University, Ra'anana, Israel
3 tamirta@openu.ac.il

***Abstract*—Analyzing the structure of social networks is of interest in a wide range of disciplines, but such activity is limited by the fact that these data represent sensitive information and can not be published in their raw form. One of the approaches to sanitize network data is to randomly add or remove edges from the graph. Recent studies have quantified the level of anonymity that is obtained by random perturbation by means of a-posteriori belief probabilities and, by conducting experiments on small datasets, arrived at the conclusion that random perturbation can not achieve meaningful levels of anonymity without deteriorating the graph features.**

We offer a new information-theoretic perspective on this issue. We make an essential distinction between image and preimage anonymity and propose a more accurate quantification, based on entropy, of the anonymity level that is provided by the perturbed network. We explain why the entropy-based quantification, which is global, is more adequate than the previously used local quantification based on a-posteriori belief. We also prove that the anonymity level quantified by means of entropy is always greater than or equal to the one based on a-posteriori belief probabilities. In addition, we introduce and explore the method of random sparsification, which randomly removes edges, without adding new ones.

Extensive experimentation on several very large datasets shows that randomization techniques for identity obfuscation are back in the game, as they may achieve meaningful levels of anonymity while still preserving features of the original graph.

I. Introduction

Social networks are graph structures holding information on sets of entities and the relations between them. Such information is of interest in a wide range of disciplines, including sociology, psychology, market research, and epidemiology. Very often social network data cannot be published in their raw form since they might contain sensitive information. The immediate first step to respect the privacy of individuals is to remove identifying attributes like names or social security numbers from the data. However, such a naïve anonymization is far from being sufficient. As shown by Backstrom et al. [1], the mere structure of the released graph may reveal the identity of the individuals behind some of the vertices. Hence, one needs to apply a more substantial procedure of sanitization on the graph before its release.

The methods for identity obfuscation in graphs fall into three main categories. The methods in the first category provide k-anonymity in the graph via edge additions or deletions [2], [3], [4]. In the second category, methods add noise to the data in the form of random additions, deletions or switching of edges, in order to prevent adversaries from identifying their target in the network, or from inferring the existence of links between vertices [5], [6], [7], [8], [9]. The methods in the third category do not alter the graph data like the methods of the two previous categories; instead, they group together vertices into super-vertices of size at least k, where k is the required threshold of anonymity, and then publish the graph data in that coarser resolution [10], [11], [12].

In this paper we focus on the second category: changing the graph structure via random perturbations. Algorithms in this category usually utilize one of two graph perturbation strategies: random addition and deletion of edges, or random switching of edges. In the first strategy one adds randomly h non-existing edges after randomly deleting h existing edges; such techniques preserve the total number of edges in the graph [6], [7]. In the second strategy, one selects h quadruples of vertices $\{u, v, x, y\}$ where (u, v) and (x, y) are edges and (u, y) and (v, x) are not, and switches between them, so that the two former edges become non-edges and the two latter non-edges become edges [7], [8], [9]; such techniques preserve the degree of all vertices in the graph.

Hay et al. [6] investigated methods of random perturbation in order to achieve identity obfuscation in graphs. They concentrated on re-identification of vertices by their degree. Given a vertex v in the real network, they quantified the level of anonymity that is provided for v by the perturbed graph as $(\max_u\{\Pr(v|u)\})^{-1}$, where the maximum is taken over all vertices u in the released graph and $\Pr(v|u)$ stands for the belief probability that u is in fact the target vertex v. By performing experimentation on the Enron dataset, using various values of h (the number of added and removed edges), they found out that in order to achieve a meaningful level of anonymity for the vertices in the graph, h has to be tuned so high that the resulting features of the perturbed graph no longer reflect those of the original graph. Those methods were revisited in [13], in which Ying et al. compared this perturbation method to the method of k-degree anonymity due to Liu and Terzi [2]. They too used the a-posteriori belief probabilities to quantify the level of anonymity. Based on experimentation on two modestly sized datasets (Enron and

Polblogs) they arrived at the conclusion that the deterministic approach of k-degree anonymity preserves the graph features better for given levels of anonymity.

Our contributions. We present a new information-theoretic look on the strategy of random additions and deletions of edges in graphs. Our main contribution is showing that randomization techniques for identity obfuscation are back in the game, as they may achieve meaningful levels of obfuscation while still preserving characteristics of the original graph. We prove our claim by means of a principled theoretical analysis and a thorough experimental assessment.

In particular, we introduce a fundamental distinction between two different forms of privacy and the corresponding two measures of the level of anonymity achieved by the perturbed network. One measure is called k-obfuscation, or *k-image obfuscation*, and it is a measure of privacy against an adversary who tries to locate in the perturbed graph the image of a specific individual. The second is called *k-preimage obfuscation*; this is a measure of privacy against an adversary who does not have any particular target individual, but she tries to examine the released graph and deduce the identity of any of the vertices in that graph.

Our measures are defined by means of the entropy of the probability distributions that are induced on the vertices of the perturbed graph (in the case of k-obfuscation) or those which are induced on the vertices of the original graph (in the case of k-preimage obfuscation). This is in contrast to Hay et al. [6] who based their definition of k-candidate anonymity on a-posteriori belief probabilities. While the a-posteriori belief probability is a *local* measure that examines, for each vertex in the perturbed graph, the probability that the vertex originated from the target individual in question, the entropy is a *global* measure that examines the entire distribution of those belief probabilities. We explain and exemplify why the entropy-based measure is more accurate than the a-posteriori belief probability, where accuracy means that it distinguishes between situations that the other measure perceives as equivalent. Moreover, we prove that the obfuscation level quantified by means of the entropy is always no less than the one based on a-posteriori belief probabilities. We derive formulas to compute those entropies in the case where the background knowledge of the adversary is the degree of the target individual.

We conduct thorough experimentation on three very large datasets and measure multitude of features of the perturbed graph. We compare the distribution of the obfuscation levels as measured by our entropy-based measure to those measured by the previous one, and show that one may achieve meaningful levels of obfuscation while preserving most of the features of the original graph.

We also introduce the method of random sparsification, which only removes edges from the graph, without adding new ones. We compare it to random perturbation in terms of the trade-off between utility of the perturbed graphs and the levels of obfuscation that they provide. Somehow surprisingly, sparsification maintains better the characteristics of the graph than perturbation at the same anonymity levels. Indeed, removing an edge affects the structure of the graph to a smaller degree than adding an edge. This is partially due to the small-world phenomenon: adding random long-haul edges brings everyone closer, while removing an edge does not bring vertices so much apart as there usually exist alternative paths.

II. Preliminaries

Let $G = (V, E)$ be a simple undirected graph that represents some social network, i.e., each vertex v in V represents an individual and an edge (v, v') in E represents some relation between v and v'. The goal is to release the graph for the sake of public scrutiny while preserving the anonymity of the individuals, in the sense of limiting the ability of an adversary to re-identify vertices in the released graph.

A. Adversarial assumptions

When studying anonymity methods for preventing identity disclosure, it is commonly assumed that the adversary knows some structural property of the vertex representing the target individual in the real graph. Hence, if the adversary wishes to locate her target individual, Alice, in the anonymized graph, the adversary may use her prior knowledge of that structural property in order to do so. Anonymization methods aim at modifying the original graph to an anonymized graph in which the assumed property induces equivalence classes of size at least k, where k is the required level of anonymity.

Liu and Terzi [2] considered the case where the property that the adversary uses is the degree $d(v)$ of the vertex v. Namely, it is assumed that the adversary knows the degree of Alice, and armed with that knowledge the adversary embarks on a search for her in the network. The algorithms presented in [2] use edge additions, and possibly also deletions, in order to make the graph k-degree anonymous in the sense that every vertex in the graph has at least $k-1$ other vertices with the same degree. Those algorithms attempt at achieving that type of anonymity with the minimal number of edge additions and deletions.

Zhou and Pei [4] assumed a stronger property; they considered the case where the adversary knows the distance-1 neighborhood of the target vertex, $N(v_i)$, namely, the subgraph that is induced by v_i and its immediate neighbors in G. Another enhancement of the degree property appeared in [11] and [14]. They considered a sequence of properties that could be used by the adversary. $\mathcal{H}_1(v)$ is the degree of v. Then, if $t = \mathcal{H}_1(v)$, and $v_1, \ldots, v_t$ are the neighbors of v, they define $\mathcal{H}_{i+1}(v) = \{\mathcal{H}_i(v_1), \ldots, \mathcal{H}_i(v_t)\}$ for all $i \geq 1$. So, while $\mathcal{H}_1$ is just the degree property, $\mathcal{H}_2$ is the property that consists of the degrees of the neighbors, $\mathcal{H}_3$ consists of the degrees of the neighbors of the neighbors, and so forth.

In [3] it is assumed that the adversary knows all of the graph G, and the location of v_i in G; hence, she can always identify v_i in any copy of the graph, unless the graph has other vertices that are automorphically-equivalent to v_i. The algorithm that they present makes sure that every vertex in the released graph has at least $k-1$ other vertices that are automorphically-equivalent to it.

As it is very unrealistic to assume that the adversary would gain information on all neighbors of all neighbors of the target vertex, properties $\mathcal{H}_i$ for all $i \geq 2$, and the property that was assumed in [3] (which is stronger than $\mathcal{H}_i$ for all $i \geq 1$) are of more theoretical nature. But even the simpler properties of the degree, $d(v) = \mathcal{H}_1(v)$, and the neighborhood, $N(v)$, are quite strong, since typically an adversary would be able to gain information on neighbors of Alice and their interconnection, but it would be hard for him to get hold of the complete picture. Hence, a more realistic assumption is that the adversary knows a partial picture regarding $N(v_i)$. For instance, the adversary can be a member of the same hiking group as Alice's. If that group has 20 members, all of whom know each other since they meet every other weekend, the adversary would know that Alice's neighborhood contains a clique of size 20. In such cases the property is that of a sub-neighborhood. The adversary knows that the neighborhood of the target vertex, $N(v_i)$, contains some graph H. With that information, she may scan all vertices in the released graph and look for those that have a neighborhood in which H may be embedded.

B. Anonymization

Let $G = (V, E)$ be a graph and P be a property of the vertices in the graph, as exemplified above. That property induces an equivalence relation $\sim$ on V, where $v_i \sim v_j$ if and only if $P(v_i) = P(v_j)$. For example, for the degree property, $v_i \sim v_j$ if $d(v_i) = d(v_j)$; or, in the case that was studied in [3], $v_i \sim v_j$ if they are automorphically-equivalent. Since P is assumed to hold all of the information that the adversary has regarding the target vertex, she cannot distinguish between two vertices in the same P-equivalence class. This motivates the following definition.

Definition 2.1: A graph G is called k-anonymous with respect to property P if all equivalence classes in the quotient set $V/\sim$ of the property P are of size at least k.

Given an integer k, one may transform the input graph G into a k-anonymous graph $G_A = (V_A, E_A)$ by adding or removing edges and vertices from G. For example, in the model of k-degree anonymity [2], the anonymization algorithm adds edges to the graph (or, in another version of the algorithm, adds and removes edges) until each degree in $G_A = (V_A = V, E_A)$ appears at least k times. In k-neighborhood anonymity [4], edges are added until each neighborhood in G_A appears at least k times. In k-symmetry anonymity [3], new edges and vertices are added until every vertex in G_A has at least $k-1$ other vertices that are indistinguishable from it based on the graph structure, since they are all automorphically-equivalent.

By releasing a graph that was subjected to such a k-type anonymity procedure, the adversary will not be able to track Alice down to subsets of vertices of cardinality less than k. Moreover, as all vertices in that set are equivalent in the eyes of the adversary (since they all share the same property which is the only weapon that the adversary has), they are all equally probable as being Alice. Hence, if we model the knowledge of the adversary regarding the location of Alice in V_A as a random distribution on V_A, where each vertex v in V_A has an associated probability of being Alice, given the a-priori knowledge of the adversary and the observed G_A, then k-type anonymity models dictate that the entropy of that random distribution is at least $\log_2 k$.

We proceed to propose a probabilistic version of that type of k-anonymity.

III. Obfuscation by randomization

In this paper we study identity obfuscation methods based on either random perturbation or random sparsification.

A. Obfuscation by random sparsification

Obfuscation by random sparsification is performed in the following manner. The data owner selects a probability $p \in [0, 1]$ in a way that will be discussed later on. Then, for each edge e in E the data owner performs an independent Bernoulli trial, $B_e \sim B(1, p)$. He will leave the edge in the graph in case of success (i.e., $B_e = 1$) and remove it otherwise ($B_e = 0$).

Letting $E_p = \{e \in E \mid B_e = 1\}$ be the subset of edges that passed this selection process, the data owner will release the subgraph $G_p = (U = V, E_p)$. The idea is that such a graph offers some level of identity obfuscation for the individuals in the underlying population, while maintaining sufficient utility in the sense that many features of the original graph may be inferred from looking at G_p.

The set of vertices in G_p will be denoted by U, even though it equals the set of vertices in G, which is denoted V. The introduction of a different notation will be needed in our analysis later on, in order to distinguish between the set of vertices, as observed by the adversary in G_p, and the set of vertices in the original graph, G.

B. Obfuscation by random perturbation

Obfuscation by random perturbation is a process that consists of two phases – edge deletions followed by edge additions. One way of performing this process is as follows: In the first phase the data owner selects an integer h and then he randomly picks a subset of h edges out of the m edges in E and removes them. In the second phase the data owner randomly picks h pairs of vertices that were not connected in E, and adds edges that connect them.

We consider here random perturbations that use a very similar process using a sequence of Bernoulli trials. In the first phase the data owner selects a probability $p \in [0, 1]$ and then, for each edge e in E, he retains it in probability p. In the second phase the data owner selects another probability q and then adds an edge e in $(V \times V) \setminus E$ with probability q. In order to guarantee that the expected number of edges in the resulting graph equals $m = |E|$, the probability q should be selected so that

$$pm + q \cdot \left(\binom{n}{2} - m \right) = m,$$

or equivalently,

$$q = q(p) = \frac{m}{\left(\binom{n}{2} - m\right)} \cdot (1-p). \qquad (1)$$

As we shall always set q to be a function of p through Equation (1), we shall denote the resulting randomized graph, as before, by $G_p = (U = V, E_p)$.

IV. k-Obfuscation and k-preimage obfuscation

Here, we define our two privacy notions. The first one protects against adversaries who try to locate a specific individual in the randomized graph. The second one protects against a different type of adversarial attack which is not targeted against a specific individual. (A similar distinction was made in [15] between $(1, k)$-anonymity and $(k, 1)$-anonymity in the context of anonymizing databases by means of generalization.)

A. *k-Obfuscation*

We assume hereinafter that the adversary knows the randomization method and the value of the selected randomization parameter p. The goal of the adversary is to locate the image in U of a specific vertex v in V. Due to randomization, the adversary cannot determine which of the vertices u in U is the image of v in V; however, based on her background knowledge and the observed G_p the adversary may associate a probability with every vertex u in U as being the sought-after image of v in V. Let us denote the corresponding random variable that is defined by v and G_p on U by X_v; namely, for every u in U, $X_v(u)$ is the probability that u is the image of v in G_p.

Definition 4.1 (k-Obfuscation): A perturbed graph G_p respects k-obfuscation if for every vertex v in V, the entropy of the random variable X_v over U is at least $\log k$.

We used the term obfuscation, rather than anonymity, because traditionally anonymity is associated with cases in which every item (vertex or record) in the released data (graph or table) belongs to an equivalence class of items of size at least k. Randomization does not produce such outputs, hence the different term.

Hay et al. [11] defined a different notion: k-candidate anonymity. Reformulated in our terms, a randomized graph offers k-candidate anonymity if

$$X_v(u) \le \frac{1}{k}, \qquad \text{for all } v \text{ in } V \text{ and } u \text{ in } U. \qquad (2)$$

The logic behind that definition, as implied by the term *k-candidate* anonymity, is that condition (2) guarantees that for each vertex $v \in V$ there are at least k candidate nodes $u \in U$ that might be its image. Hence, k-candidate anonymity attempts at guaranteeing a lower bound on the amount of uncertainty that the adversary would have when she tries to locate the image of the target individual in the perturbed graph. However, we claim that this definition does not measure correctly the amount of uncertainty that the adversary has regarding the correct identification of the target individual. For example, such a definition does not distinguish between the following two situations:

(1) $X_v(u_1) = X_v(u_2) = \frac{1}{2}$, and
$X_v(u_i) = 0$ for all $3 \le i \le n$;

(2) $X_v(u_1) = \frac{1}{2}$,
$X_v(u_i) = \frac{1}{2t}$ for all $2 \le i \le t+1$, and
$X_v(u_i) = 0$ for all $t+2 \le i \le n$.

Both cases respect 2-candidate anonymity since the maximal probability in both is $\frac{1}{2}$. However, it is clear that in the first case, where there are only two suspects, the amount of uncertainty (and the efforts that are needed to complete the identification) is much smaller than in the second case, where there are $t+1$ suspects.

The correct way to measure uncertainty is the entropy. Indeed, the entropy distinguishes between the above two cases: By applying Definition 4.1 to those two cases, we find that the first one respects 2-obfuscation, while the second one respects $(2\sqrt{t})$-obfuscation.

It is worth noting that ℓ-diversity [16] is a measure of exactly the same thing — the amount of uncertainty of the adversary regarding some property of the target individual. There too, the selected measure is the entropy and not the maximal probability. In other studies that used ℓ-diversity, the interpretation of ℓ-diversity was taken as the maximal probability instead of the entropy, e.g. [17], [18], [19]. However, the maximal probability was taken instead of the entropy just because it is easier to enforce and not because it was perceived as a better measure for the diversity.

Proposition 4.2: The obfuscation level of a given perturbed graph G_p is always no less than the corresponding candidate anonymity level.

Proof: Fix $v \in V$ and let $\mathbf{p}(v) = (p_1, \ldots, p_n)$ denote the probability distribution X_v; i.e., if the vertices in the perturbed graph are $U = \{u_1, \ldots, u_n\}$ then $p_i = X_v(u_i)$. The obfuscation level offered by G_p is then

$$k_o = \min_{v \in V} 2^{H(\mathbf{p}(v))}$$

while the candidate anonymity level is

$$k_c = \min_{v \in V} \left(\max_{p_i \in \mathbf{p}(v)} p_i \right)^{-1}.$$

For any fixed $v \in V$ we have

$$H(\mathbf{p}(v)) = \sum_i p_i \log\left(\frac{1}{p_i}\right) \ge \sum_i p_i \log\left(\frac{1}{\max p_i}\right)$$

$$= \log\left(\frac{1}{\max_{p_i \in \mathbf{p}(v)} p_i}\right).$$

Therefore,

$$2^{H(\mathbf{p}(v))} \ge \left(\max_{p_i \in \mathbf{p}(v)} p_i\right)^{-1} \quad \forall v \in V.$$

The last inequality implies that $k_o \ge k_c$. ■

In experiments that were conducted in [11] on the Enron dataset, it was shown that in order to achieve a reasonable k-candidate anonymity it is necessary to use values of the perturbation probability p for which most of the graph features (e.g., diameter, path length, closeness, betweenness) are severely altered. They concluded that randomness cannot achieve at the same time a reasonable privacy preservation and an acceptable loss of utility. Our claim here is that, in light of the observation that k-obfuscation is the correct measure, and that such a measure is satisfied by larger values of k than k-candidate anonymity, random perturbation is back in the game of privacy preservation in social networks.

B. k-preimage obfuscation

The definition of k-obfuscation and k-candidate anonymity reflect the goal to protect against an adversary who wishes to reveal sensitive information on a specific target individual. Those definitions aim to limit the possibility of the adversary to identify the image of the target individual in the released network. Such adversarial attacks are the ones that are more commonly discussed and analyzed, e.g. [1], [20].

However, there is another type of adversarial attacks that one should consider: When the adversary is interested in re-identifying any entity in the released data, for instance to find possible victims to blackmail. Such an attack works in the opposite direction: Focusing on an item in the released (and presumably anonymized) corpus of data, the adversary tries to infer its correct preimage.

This kind of attack is also at the basis of the well-known August 2006 AOL crisis. On August 4, 2006, AOL Research released a compressed text file containing twenty million search keywords for more than 650,000 users over a 3-month period, intended for research purposes. Even though all records were stripped of the identity of the person that made the query, certain keywords contained personally identifiable information. Since each user was identified on that list by a unique sequential key, it was possible to compile a search history for each given user. The New York Times [21] was able to locate an individual from the released and anonymized search records by cross referencing them with phonebook listings.

These considerations motivate the following definition.

Definition 4.3 (k-Preimage Obfuscation): Let $G = (V, E)$ and $G_p = (U, E_p)$ be an original and perturbed graphs, respectively. For each $u \in U$ let X_u denote the corresponding random variable that is defined on V, i.e., $X_u(v)$ is the probability that v is the preimage of u in G. Then the perturbed graph G_p respects k-preimage obfuscation if for every vertex $u \in U$, the entropy of the random variable X_u on V is at least $\log k$.

Similarly, we may define k-preimage candidate anonymity by enforcing $X_u(v) \leq \frac{1}{k}$ for all $v \in V$ and $u \in U$.

V. Quantifying the level of obfuscation

The definitions in the previous section involved two ensembles of probability distributions: $\{X_v(\cdot) : v \in V\}$, defined on U, for k-obfuscation, and $\{X_u(\cdot) : u \in U\}$, defined on V, for k-preimage obfuscation. The randomized graph G_p respects k-obfuscation (resp. k-preimage obfuscation) if the entropy of each of the probability distributions in the first (resp. second) ensemble is greater than or equal to $\log k$. Here we describe how to compute those distributions. We separate the discussion to the two notions of obfuscation.

A. Verifying k-obfuscation

Let $P(\cdot)$ denote the property that the adversary knows about the target vertex $v \in V$. By looking at the released graph G_p and the values of $P(u_i)$ for each of the vertices $u_i \in U$ in G_p, the adversary may associate a probability $X_v(u)$ for the event that the vertex $u \in U$ is the image of v.

Let v and u be vertices in V and U respectively. Let $f(v; u)$ be the probability that a vertex with property $P(v)$ was converted, under the known randomization model, to a vertex with property $P(u)$. For example, in the case of the degree property, $P(\cdot) = d(\cdot)$, $f(v; u)$ is the probability that a vertex with degree $d(v)$ was converted to a vertex with degree $d(u)$, given the method and parameters of randomization. For the sake of illustration, if the method of randomization was sparsification, then $f(v; u) = 0$ whenever $d(v) < d(u)$, since by only deleting edges it is impossible that the degree of a vertex would increase.

As discussed earlier, the property P induces an equivalence relation, denoted $\sim$, on both V and U. Therefore, we may compute the probabilities $f(v; u)$ only on the Cartesian product of the two equivalence classes, $(V/\sim) \times (U/\sim)$. Those computed values would give the values of $f(v; u)$ for all $v \in V$ and $u \in U$. We arrange those values in an $n \times n$ matrix F where $F_{i,j} = f(v_i, u_j)$, $1 \leq i, j \leq n$. Each row in this matrix corresponds to an original vertex $v \in V$ and gives the related probabilities $f(v, u)$ for all $u \in U$. Similarly, each column corresponds to an image vertex $u \in U$. The matrix F enables us to compute the probability distributions $X_v(\cdot)$, for all $v \in V$, by looking at its rows. The probability distributions are obtained by normalizing the corresponding row in the matrix F:

$$X_{v_i}(u_j) = \frac{F_{i,j}}{\sum_{1 \leq j \leq n} F_{i,j}}, \quad 1 \leq i, j \leq n. \tag{3}$$

For example, if in the randomization process we use $p = 1$, then $G_p = G$. In that case, $f(v; u) = 1$ if $P(v) = P(u)$ and $f(v; u) = 0$ otherwise. Therefore, for any given $v \in V$, the entries in the corresponding row in F would be 1 in columns that correspond to vertices $u \in U$ with $P(u) = P(v)$, and 0 otherwise, since by using $p = 1$ (no randomization), the properties of the vertices remain unchanged. For each $v \in V$, the set of probabilities $\{f(v; u) : u \in U\}$ is then converted into a probability distribution by means of normalization. If there are ℓ vertices $u \in U$ for which $f(v; u) = 1$, then each one of them is the image of v with probability $1/\ell$. In that case, $X_v(\cdot)$ associates the probability $1/\ell$ for each of those ℓ vertices in U, and zero to the remaining $n - \ell$ vertices.

To illustrate that, assume that G and G_p have 7 vertices and $p = 1$ (so that $G_p = G$). Assume that the degree sequence in the graph is $(2, 2, 2, 3, 4, 4, 5)$. Then Table I below gives the matrix F is this case; the first column indicates the vertices in G and their degrees, while the first row indicates the vertices in G_p and their degree. Take, for instance, the row of v_1. It has four 0s and three 1s. Then by dividing the entries of that row by 3, we infer that each of the vertices u_1, u_2, u_3 is the image of v_1 with probability $1/3$, while u_4, u_5, u_6, u_7 cannot be the image of v_1.

TABLE I	u_1:2	u_2:2	u_3:2	u_4:3	u_5:4	u_6:4	u_7:5
v_1:2	1	1	1	0	0	0	0
v_2:2	1	1	1	0	0	0	0
v_3:2	1	1	1	0	0	0	0
v_4:3	0	0	0	1	0	0	0
v_5:4	0	0	0	0	1	1	0
v_6:4	0	0	0	0	1	1	0
v_7:5	0	0	0	0	0	0	1

The computation of the probability distributions $X_v(\cdot)$ that was carried out in [6] was different, but the basic idea is similar. In that study, the randomization was made by first removing h of the existing edges, thus arriving at an interim graph, and then adding h of the edges that do not exist in the interim graph, thus arriving at G_p. Given a graph G_p, the set of possible worlds $\mathcal{W}_h(G_p)$ is the collection of all graphs over the same set of vertices that could have been the pre-image of G_p under that randomization model. Each of the graphs G in $\mathcal{W}_h(G_p)$ was associated with a probability, $\Pr(G)$. Then, they defined $f(v; u)$ as the sum of $\Pr(G)$ for all possible worlds $G \in \mathcal{W}_h(G_p)$ in which the property of the candidate vertex u equals the known property of the target vertex v. Finally, they normalized $f(v; u)$ in the same way that we did in order to get a probability distribution $X_v(\cdot)$ on U. In particular, the probability distribution that was associated in [6] to every possible world was uniform. However, this should be corrected since some possible graphs may be converted to the observed randomized graph G_p in more than one way. Specifically, if the intersection of the edge set in the possible graph G with the edge set in G_p is of size $m-h+j$, where $0 \leq j \leq h$, then the number of different ways in which G could be converted to G_p is $\binom{m-h+j}{j}$. As a result, the probability of the possible world G is

$$\Pr(G) = \binom{m-h+j}{j} \Big/ \left[\binom{m}{h} \binom{\binom{n}{2} - m + h}{h} \right].$$

B. Verifying k-preimage obfuscation

Assume that the adversary knows the value of the property $P(\cdot)$ for all vertices $v \in V$. Then by looking at the released graph G_p and fixing $u \in U$, he may associate, for every $v \in V$, a probability $X_u(v)$ for the event that v is the preimage of u. Namely, every $u \in U$ induces a probability distribution on V.

Let u and v be vertices in U and V respectively. Let $f'(v; u)$ be the probability that a vertex with property $P(u)$ originated from a vertex with property $P(v)$. As discussed earlier, we may compute $f'(v; u)$ on the Cartesian product of the two equivalence classes, $(V/\sim) \times (U/\sim)$, and then construct an $n \times n$ matrix F' where $F'_{i,j} = f'(v_i, u_j)$, $1 \leq i, j \leq n$. That matrix enables us to compute the probability distributions $X_u(\cdot)$, for all $u \in U$, by looking at its columns. The probability distributions are obtained by normalizing the corresponding column in the matrix F':

$$X_{u_j}(v_i) = \frac{F'_{i,j}}{\sum_{1 \leq i \leq n} F'_{i,j}}, \quad 1 \leq i, j \leq n. \tag{4}$$

Considering the example that was given above in Section V-A, if we normalize the column of u_5 in Table I we infer that u_5 originated from v_5 or v_6 with probability $1/2$ each, and could not have originated from any of the other vertices.

We proceed to derive explicit formulas for the case where the property is the degree. The goal is to arrive at a condition that p has to satisfy so that G_p will be k-obfuscated. In Section VII we discuss stronger properties.

VI. k-DEGREE OBFUSCATION

Here we consider the case of the degree property $P(\cdot) = d(\cdot)$ and derive explicit formulas for $f(\cdot;\cdot)$ and $f'(\cdot;\cdot)$. From those values one may compute the levels of k-degree obfuscation and k-degree preimage obfuscation as described in the previous section. Hereinafter, if $v \in V$ and $u \in U$, and u is the image of v, we denote this relation by $v \mapsto u$.

Let $v \in V$ be a target vertex in V and assume that the adversary knows that its degree is $d(v) = a$. Let $u \in U$ be a candidate vertex in U whose degree is $d(u) = b$. Then $f(v; u)$ equals the following conditional probability,

$$f(v; u) = \Pr(d(u) = b \mid d(v) = a, v \mapsto u); \tag{5}$$

namely, given that v has a degree a and its image in G_p is u, $f(v; u)$ is the probability that u's degree is b. (In order to avoid cumbersome notations we shall drop the notation $v \mapsto u$ from the conditional probabilities henceforth; it is assumed always that v and u are a preimage and image pair.)

Under the random sparsification approach, $b \sim B(a, p)$, where $B(a, p)$ is the Binomial distribution over a experiments and success probability p. Under the random perturbation approach, $b \sim B(a, p) + B(n-1-a, q(p))$. Hence, in the first model of random sparsification,

$$\Pr(d(u) = b \mid d(v) = a) = \begin{cases} \binom{a}{b} p^b (1-p)^{a-b} & b \leq a \\ 0 & b > a \end{cases}. \tag{6}$$

As for the second model of random perturbation, let us denote by t the number of real edges adjacent to v that survived the randomization. Hence, $b - t$ is the number of phantom edges that were added by the random perturbation. Clearly, $t \leq a$ and $t \leq b$. Therefore, the conditional probability is given by

$$\Pr(d(u) = b \mid d(v) = a) = \sum_{t=0}^{\min\{a,b\}} \binom{a}{t} p^t (1-p)^{a-t} \times \\ \times \binom{n-1-a}{b-t} q^{b-t} (1-q)^{n-1-a-b+t}. \tag{7}$$

Let

$$F(v) = \sum_{u \in U} f(v; u), \tag{8}$$

where $f(v;u)$ is given by Equations (5)+(6) or (5)+(7), and for every $u \in U$, b is its degree. Then

$$X_v(u) = \frac{f(v;u)}{F(v)}, \quad u \in U.$$

Hence, the level of obfuscation which is provided by using p is $k_o = \min_{v \in V} 2^{H(X_v)}$, while the candidate anonymity level is $k_c = \min_{v \in V} \left(\max X_v\right)^{-1}$.

Next, we turn to compute $f'(v;u)$, which is the inverse conditional probability. Assume that $d(v) = a$ and $d(u) = b$. Then

$$f'(v;u) = \Pr(d(v) = a \mid d(u) = b); \tag{9}$$

namely, given that v is the preimage of u and that the degree of u in G_p is b, $f'(v;u)$ is the probability that the degree of v is a. By Bayes Theorem,

$$\begin{aligned} &\Pr(d(v) = a | d(u) = b) = \\ &= \frac{\Pr(d(v) = a)}{\Pr(d(u) = b)} \cdot \Pr(d(u) = b | d(v) = a). \end{aligned} \tag{10}$$

The value of $\Pr(d(v) = a)$ may be obtained from the frequencies of the degrees in G. The probabilities $\Pr(d(u) = b)$ are computed as follows:

$$\Pr(d(u) = b) = \sum_a \Pr(d(u) = b | d(v) = a) \cdot \Pr(d(v) = a). \tag{11}$$

Hence, the value of $f'(v;u)$ is given by Equations (9)–(11), together with Equations (6) or (7) which give the conditional probability $\Pr(d(u) = b | d(v) = a)$ in the two randomization models.

To summarize, we derived here the values of $f(v;u)$ and $f'(v;u)$ in the case of $P(\cdot) = d(\cdot)$. Those values enable to compute the probability distributions $X_v(\cdot)$ on U, for all $v \in V$, and $X_u(\cdot)$ on V, for all $u \in U$. The data owner needs to select a randomization parameter p such that the entropy of X_v is greater than or equal to $\log k$ for all $v \in V$ (in order to satisfy k-obfuscation), or a similar condition for the X_u, $u \in U$, probability distributions on V for the sake of k-preimage obfuscation. The goal is to select the largest p for which the required level of obfuscation is met, in order to retain as much as possible of the properties of the original graph. The maximal value of p that still respects k-obfuscation (or k-preimage obfuscation) may be approximated by numerical means.

VII. k-NEIGHBORHOOD OBFUSCATION

Here we discuss briefly the case in which the traceability property is the neighborhood, $P(\cdot) = N(\cdot)$. In order to compute the probability distributions $X_v(\cdot)$, $v \in V$, on U, in this case, we need to compute, for every pair of neighborhoods α and β, the conditional probability $\Pr(N(u) = \beta \mid N(v) = \alpha,\, v \mapsto u)$. This is the probability that a vertex $v \in V$ that has a neighborhood α is converted under the randomization model to a vertex $u \in U$ with a neighborhood β. In the case of randomization by sparsification, it is necessary to find all possible embeddings of β in α, since there could be many ways in which α could be transformed into β, to compute the probability of each such transformation and add them up. Such a computation seems intricate even for moderately sized α and β. The situation becomes more intricate in the case of random perturbation. Here, any neighborhood α could be converted to any neighborhood β since any edge can be potentially removed and any non-edge (in the entire graph) can be potentially added.

Hence, it seems hard to measure precisely the level of obfuscation that is achieved when the property is not a simple one like the degree. The same difficulty also prevents the adversary from associating probabilities to the different vertices in the released graph as the possible images of the target vertex. Moreover, as opposed to the degree property, in order to perform such computations in the perturbation model, the adversary would need to know the structure of the entire graph, since even two far-apart vertices may become neighbors in that randomization model.

VIII. EXPERIMENTS

In this section we report our experimental assessment of the effects of random sparsification and perturbation on the structure of the perturbed graph, as well as of the level of anonymity achieved, according to both k-obfuscation and k-preimage obfuscation notions. In all of the experiments we assume an adversary that uses the degree as the re-identification property, and that knows the randomization method and the value of the randomization parameter p.

A. *Datasets*

We use three large real-world datasets – dblp, flickr, and Y360. The main characteristics of the datasets are provided in Table II, where n is the number of vertices, m is the number of edges, d is the average degree, Δ is the maximum degree, and α is the coefficient of fitting a power law in the degree distribution.

dblp. We extract a co-authorship graph from a recent snapshot of the DBLP database that considers only the journal publications. There is an undirected edge between two authors if they have coauthored a journal paper.

flickr. Flickr is a popular online community for sharing photos, with millions of users. In addition to many photo-sharing facilities, users are creating a social network by explicitly marking other users as their *contacts*. In our dataset, vertices represent users and edges represent the contact relationship.

TABLE II
DATASET CHARACTERISTICS

Dataset	n	m	d	Δ	α
dblp	226 413	716 460	6.32	238	2.8
flickr	588 166	5 801 442	19.72	6 660	1.9
Y360	1 226 311	2 618 645	4.27	258	2.3

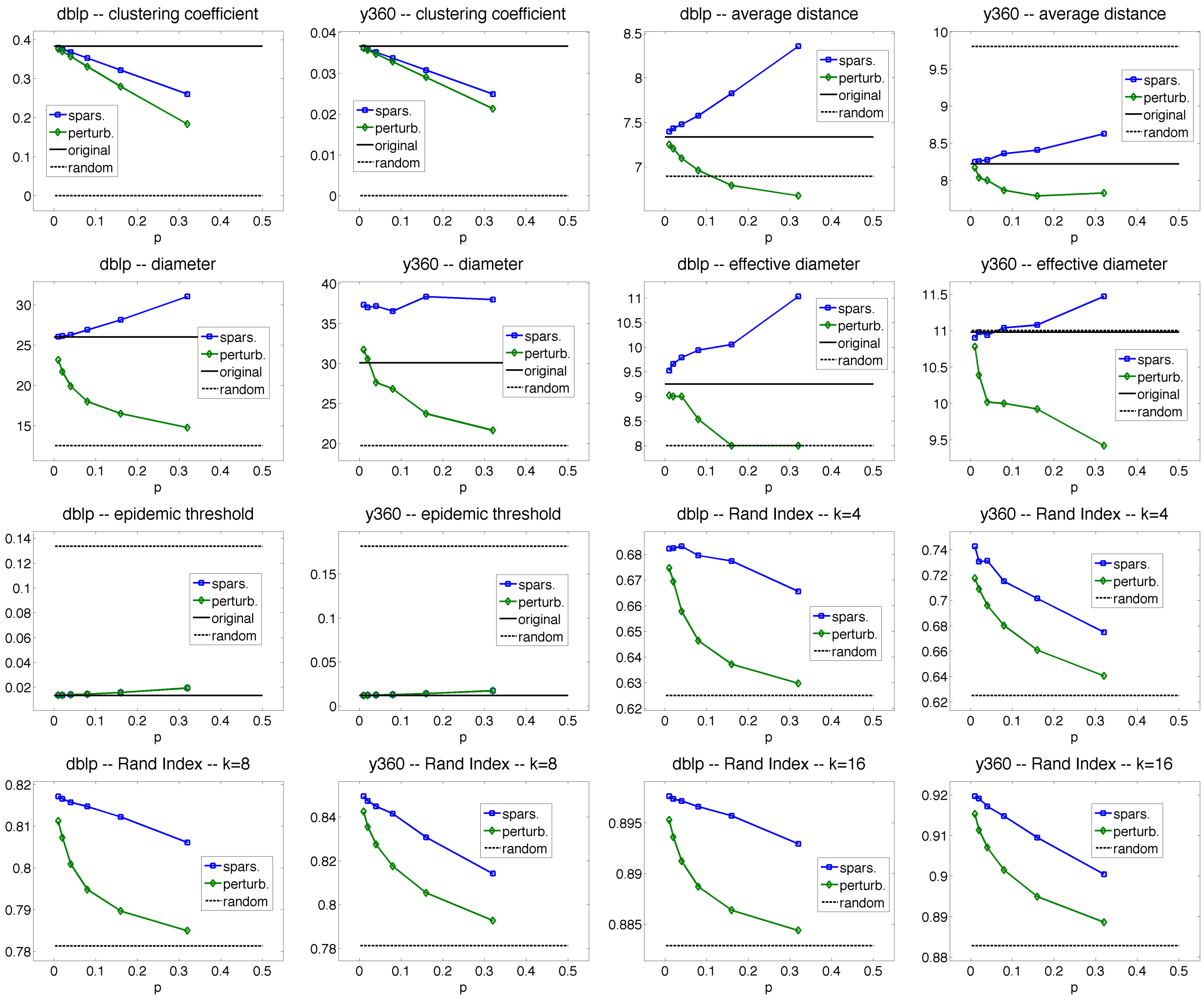

Fig. 1. Effect of randomization on the `dblp` and `Y360` datasets.

`Y360`. Yahoo! 360 was a social networking and personal communication portal. Our dataset models the friendship relationship among users. Among the three datasets, `Y360` is the sparsest one.

B. Graph statistics

The first objective of our experimental evaluation is to show that the method of randomized anonymization leaves to a large extent the structure of the original graph intact. Our strategy is to measure certain graph statistics on the original graph, on the anonymized graphs, and on random graphs. We then expect that for the anonymized graphs, the statistics under consideration are closer to those statistics on the original graph than to those on the random graph. The statistics that we measure are the *clustering coefficient*, (i.e., the fraction of closed triplets of vertices among all connected triplets), the *average distance* among pairs of vertices, the *diameter* (i.e., the maximum distance among pairs of vertices), the *effective diameter* (the 90th percentile distance, i.e., the minimal value for which 90% of the pairwise distances in the graph are no larger than), and the *epidemic threshold* (defined later).

We also report experiments based on graph clustering. We run the METIS graph-clustering algorithm [22] with a prefixed number of clusters k on the original graph and on the perturbed one and we report their *Rand Index*, which is a measure of clustering similarity.[1]

In Figure 1 we report all of the above statistics, for different values of p on the `dblp` and `Y360` datasets. We compare the graph obtained by sparsification and perturbation with a random Erdõs-Rényi graph containing the same number of edges and averaged over 50 random runs, and with the original

[1] http://en.wikipedia.org/wiki/Rand_index

graph. For the perturbation approach, the value of q is defined as a function of p according to Equation (1) in Section III. In all plots we report six values per curve, corresponding to $p = 2^i/100$ for $i = 0, \dots, 5$.

We next describe the results following the order of plots in Figure 1 from left to right and from top to bottom.

Clustering coefficient. The first two plots of Figure 1 show that in both datasets, sparsification and perturbation do not lose much in terms of clustering coefficient for small values of p. When p grows to 0.16 the loss starts to be more substantial. In both datasets, sparsification preserves better the clustering coefficient.

Average distance, diameter and effective diameter. The next six plots of Figure 1 show that sparsification obviously increases distances among pairs of vertices. Perturbation, on the other hand, drastically reduces the diameter of the network, since the addition of random long-haul edges brings everyone closer. Overall, sparsification preserves better the graph distances than perturbation does, with the exception of the diameter in `Y360`. However, it should be noted that in the effective diameter (which is a smoother and more robust to noise measure) sparsification performs very well especially for reasonably low values of p.

Epidemic threshold. An interesting way to characterize complex networks is by studying their epidemic properties. The idea here is to assume that a virus propagates along the edges of the graph, according to some virus-propagation model, and infects vertices in the graphs. It turns out that for many virus-propagation models of interest, the graph can be characterized by a quantity called *epidemic threshold*, and the epidemic exhibits a threshold phenomena: if a certain parameter of the virus propagation model exceeds the epidemic threshold, then the virus will spread to all the vertices of the graph, otherwise it will die out. In certain virus-propagation models [23], it can be shown that the epidemic threshold is λ_1^{-1}, where λ_1 is the largest eigenvalue of the adjacency matrix of the graph. This is the definition of epidemic threshold that we use here.

The values of epidemic threshold for `dblp` and `Y360` datasets are reported in the first half of the third row of Figure 1. The results are very intuitive: real graphs have very low epidemic thresholds; essentially epidemic outbreaks are likely because of the existence of hubs. On the other hand, random graphs have high tolerance to epidemic outbreaks. Perturbation and sparsification both produce graphs that have epidemic thresholds which are very close to that of the original graph.

Clustering similarity. Maybe the most interesting experiment is to asses to which extent the results of a data mining analysis are sensitive to the perturbation process. In the last two rows of Figure 1 we report the similarity measure between a clustering obtained on the original graph and a clustering of the perturbed one. We can observe that both perturbation and sparsification perform well for small values of p, being always above 0.8 similarity, but as p grows, perturbation has a larger effect on clustering similarity than sparsification.

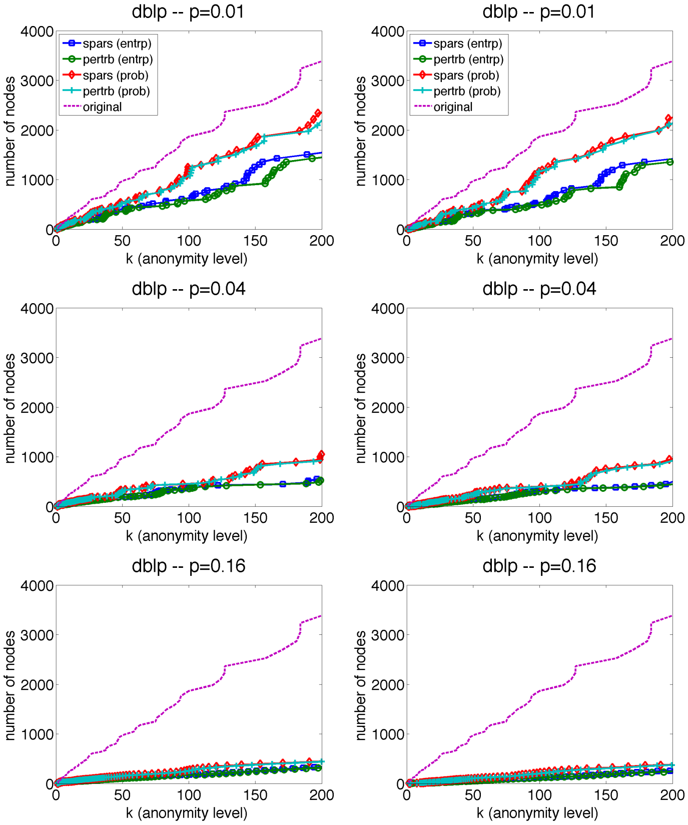

Fig. 2. Anonymization level of k-obfuscation (left) and k-preimage obfuscation (right) on the `dblp` dataset.

C. Anonymization level

So far we have discussed the extent to which the structure of the graph is preserved for some given values of p. The next question is the level of anonymity which is achieved for the same values of p. Figures 2 and 3 report this information for `dblp` and `Y360` datasets respectively. The two columns in the plots refer to the two adversarial models that we discussed. The left column reports the achieved levels of k-obfuscation and k-candidate anonymity, while the right column reports the achieved levels of k-preimage obfuscation and k-preimage candidate anonymity. Specifically, for a given value of p, we have on the x-axis the anonymity level and on the y-axis the number of vertices that *do not* reach such anonymity level. The plots report those values for the original graph, and for the sparsified and perturbed graphs. The curves that correspond to obfuscation levels are marked by (entrp) (since they are based on the entropy) while those that correspond to candidate anonymity are marked by (prob) (since they are based on the probability values).

We can draw three observations from these plots:

(1) Both perturbation and sparsification provide good privacy protection already at very low values of p.

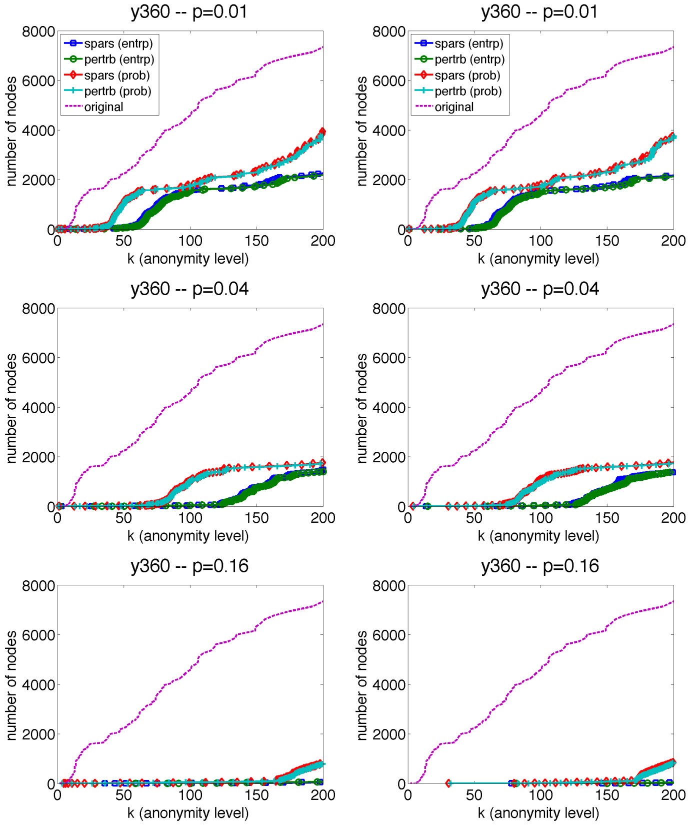

Fig. 3. Anonymization level of k-obfuscation (left) and k-preimage obfuscation (right) on the `Y360` dataset.

(2) As proved in Proposition 4.2, the obfuscation level measured by entropy is always larger than the obfuscation level measured by probability.

(3) Although, in general, given a graph G and a randomization G_p, the corresponding level of image obfuscation is usually different from the corresponding level of preimage obfuscation, it turns out that in practice the plots for k-obfuscation and k-preimage obfuscation levels tend to be very similar.

Comparison with Liu-Terzi [2]. We next present a comparison with the method for k-degree anonymity by Liu and Terzi [2] (LT hereinafter). This set of experiments is conducted on the `flickr` dataset and the results are reported in Figures 4 and 5. In all of the experiments we assume an anonymity parameter $k = 20$ for the LT method.

We recall that the objective of the LT method is to find the minimum number of edge additions and deletions so that the resulting graph is k-degree anonymous. To understand better the behavior of the LT method, we consider a typical real-world graph whose degree distribution follows a power law. We observe that for such a typical power law graph, almost all vertices satisfy already the k-degree anonymity requirement, with the possible exception of a small number of hubs. Thus, the LT algorithm has only to "adjust" the degrees of those hubs (possibly by adding or deleting edges among them) while leave the majority of the other vertices unaffected. Therefore, LT is able to achieve k-degree anonymity with a very small number of edge additions and deletions. Even though the resulting graph satisfies the k-degree anonymization requirement, one may argue that leaving the majority of the vertices unaffected has serious privacy implications. For example, the resulting graph is still vulnerable to the attack of Backstrom et al. [1].

In contrast, the randomized algorithm affects all vertices in the graph and thus it offers stronger anonymization properties. In a sense, the randomized anonymization destroys not only degree structure, but also higher neighborhood structures. Unfortunately, as we pointed out previously, quantifying the anonymization level for higher neighborhood structures seems a computationally infeasible problem (for the data owner as well as for the adversary).

Our intuition is made clearer by studying the plots in Figure 4. We observe that all vertices in the output graph of the LT method satisfy k-degree anonymity with $k = 20$; however, when taking higher values of k, the curve of anonymity levels in the LT-anonymized graph quickly converges to the curve of anonymity levels in the original graph. On the other hand, the randomization methods provide significantly better levels of anonymization for all vertices in the graph, with the exception of the very small minority of vertices whose original anonymization level is smaller than 20.

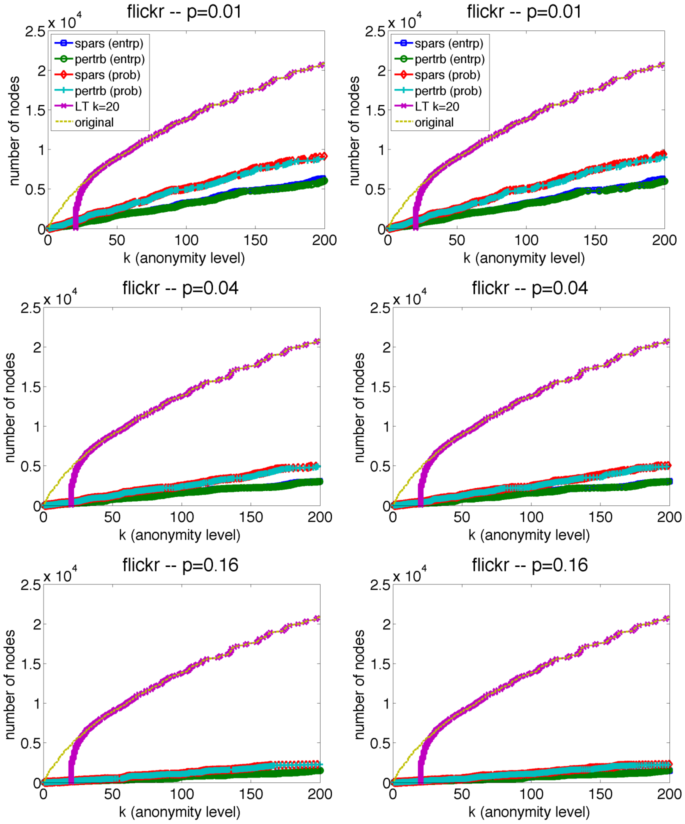

Fig. 4. Anonymization level of k-obfuscation (left) and k-preimage obfuscation (right) on the `flickr` dataset.

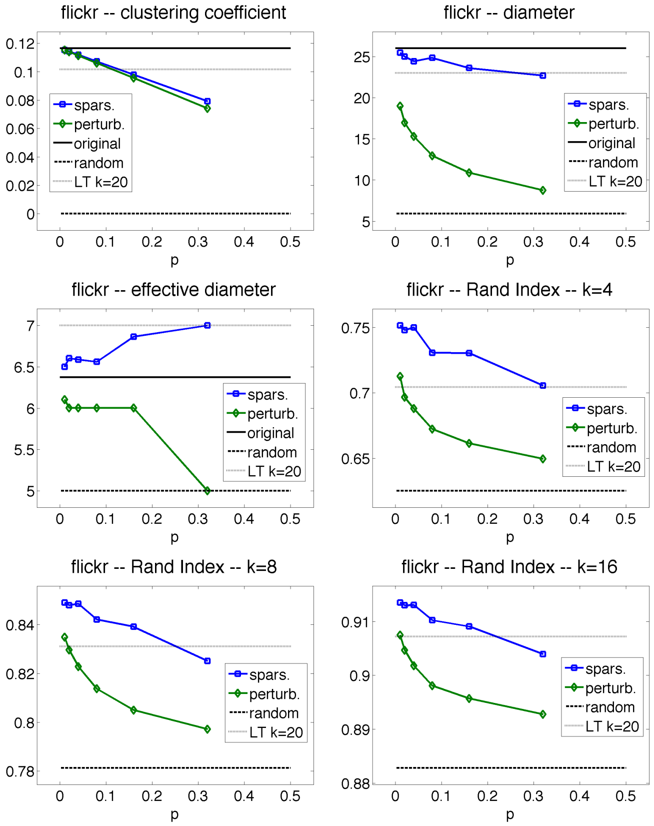

Fig. 5. Effect of randomization on the `flickr` dataset.

Nevertheless, as both LT and our randomization methods adopt the k-degree adversarial assumption, it is interesting to compare to which extent they maintain the graph structure. In Figure 5 we can observe that for reasonably small values of p, sparsification always maintains the structure of the graph better than LT with $k = 20$. For instance, for $p = 0.04$ (third point in the plots) sparsification maintains very well the graph properties, while providing enough obfuscation as reported in the second row of Figure 4.

In summary, randomization methods and the LT method have different objectives, and they operate on different ends of the anonymization spectrum, thus comparing the two methods in a fair way seems very tricky. It would be interesting to combine the two techniques; i.e., to apply first the randomization method in order to obfuscate all detectable structures in the entire graph (and not only the degrees), and then apply the LT method only for the hubs, in order to provide also for them a minimal level of degree-anonymity.

IX. Conclusions and Future Work

Randomization techniques are very appealing as privacy protection mechanisms for various reasons. They are not geared towards protecting the graph against any particular adversarial attack. They are very simple and easy to implement.

Unfortunately, recent studies have concluded that random perturbation can not achieve meaningful levels of anonymity without deteriorating the graph features. Those studies quantified the level of anonymity that is obtained by random perturbation by means of a-posteriori belief probabilities.

In this paper we offer a novel information-theoretic perspective on this issue, concluding that randomization techniques for identity obfuscation are back in the game, as they may achieve meaningful levels of obfuscation while still preserving characteristics of the original graph. We prove our claim by means of a principled theoretical analysis and a thorough experimental assessment.

We introduce an essential distinction between two different kinds of identity obfuscation, corresponding to an adversary wishing to re-identify a particular individual, or any individual. We propose to quantify the anonymity level that is provided by the perturbed network by means of entropy, and we explain why the entropy-based quantification is more adequate than the previously used quantification based on a-posteriori belief. Moreover, we prove that the obfuscation level quantified by means of the entropy is always no less than the one based on a-posteriori belief probabilities. We derive formulas to compute those entropies in the case where the background knowledge of the adversary is the degree of the target individual. We also introduce the method of random sparsification, which only removes edges from the graph.

We conduct thorough experimentation on three very large datasets and measure multitude of features of the perturbed graph, showing that randomization techniques achieve meaningful levels of obfuscation while preserving most of the features of the original graph. We also show that sparsification outperforms perturbation, as it maintains better the characteristics of the graph at the same anonymity levels.

In some settings, the network data is split between several data holders, or players. For example, the data in a network of email accounts, where two vertices are connected if they exchanged a minimal number of email messages, might be split between several email service providers. As another example, consider a transaction network where an edge denotes a financial transaction between two individuals; such a network would be split between several banks. In such settings, each player controls some of the vertices (clients) and he knows only the edges that are adjacent to the vertices under his control. It is needed to devise distributed protocols that would allow the players to arrive at a sanitized version of the unified network, without revealing to them sensitive information on clients of other players.

The recent survey by X. Wu et al. of privacy-preservation of graphs and social networks, [24, Chapter 14], concludes by recommendations for future research in this emerging area. One of the proposed directions is distributed privacy-preserving social network analysis, which “has not been well reported in literature”. The randomization method is a natural candidate method to be used in achieving anonymization in a distributed setting. It is far more suited to the distributed setting than the methods of the first and third categories that were discussed in the introduction since they hardly require a global view of the entire graph. In the full version of this paper we devise secure multi-party protocols for implementing our randomization algorithms in the distributed setting.

As pointed out in other papers (e.g., [25], [26]) the availability of additional information (e.g., interest groups) together with the published graph might create room for attacks based on *link prediction* [27]. An interesting open question is quantify the extent in which link-prediction techniques may be used as a basis for *graph reconstruction attacks*, especially against a sparsified graph.

Another interesting direction for future research is to improve the graph statistics that we obtain from the anonymized graphs. The idea is that since the anonymization is obtained with a known randomization parameter p, one may attempt to apply a correction and obtain better estimates for the clustering coefficient, distances, epidemic thresholds, and other statistics of interest.

Acknowledgements. The first two authors were partially supported by the Spanish Centre for the Development of Industrial Technology under the CENIT program, project CEN-20101037, "Social Media" (www.cenitsocialmedia.es).

Part of this research was conducted when the third author was a guest of Yahoo! Research, Barcelona.

REFERENCES

[1] L. Backstrom, C. Dwork, and J. M. Kleinberg, "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography," in *WWW*, 2007, pp. 181–190.

[2] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *SIGMOD*, 2008, pp. 93–106.

[3] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang, "k-Symmetry model for identity anonymization in social networks," in *EDBT*, 2010, pp. 111–122.

[4] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *ICDE*, 2008, pp. 506–515.

[5] S. Hanhijarvi, G. Garriga, and K. Puolamaki, "Randomization techniques for graphs," in *SDM*, 2009, pp. 780–791.

[6] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks," *University of Massachusetts Technical Report*, vol. 07, no. 19, 2007.

[7] X. Ying and X. Wu, "Randomizing social networks: A spectrum preserving approach," in *SDM*, 2008, pp. 739–750.

[8] ——, "Graph generation with prescribed feature constraints," in *SDM*, 2009, pp. 966–977.

[9] ——, "On link privacy in randomizing social networks," in *PAKDD*, 2009, pp. 28–39.

[10] A. Campan and T. Truta, "A clustering approach for data and structural anonymity in social networks," in *PinKDD*, 2008.

[11] M. Hay, G. Miklau, D. Jensen, D. F. Towsley, and P. Weis, "Resisting structural re-identification in anonymized social networks," in *PVLDB*, 2008, pp. 102–114.

[12] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationship in graph data," in *PinKDD*, 2007, pp. 153–171.

[13] X. Ying, K. Pan, X. Wu, and L. Guo, "Comparisons of randomization and k-degree anonymization schemes for privacy preserving social network publishing," in *The 3rd SNA-KDD Workshop*, 2009, pp. 1–10.

[14] B. Thompson and D. Yao, "The union-split algorithm and cluster-based anonymization of social networks," in *ASIACCS*, 2009, pp. 218–227.

[15] A. Gionis, A. Mazza, and T. Tassa, "k-Anonymization revisited," in *ICDE*, 2008, pp. 744–753.

[16] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "ℓ-Diversity: privacy beyond k-anonymity," in *ICDE*, 2006, p. 24.

[17] J. Goldberger and T. Tassa, "Efficient anonymizations with enhanced utility," *TDP*, vol. 3, pp. 149–175, 2010, a preliminary version appeared in ICDM Workshops 2009, pp. 106-113.

[18] R. Wong, J. Li, A. Fu, and K. Wang, "(α, k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing," in *KDD*, 2006, pp. 754–759.

[19] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation," in *VLDB*, 2006, pp. 139–150.

[20] R. Jones, R. Kumar, B. Pang, and A. Tomkins, "I know what you did last summer Query logs and user privacy," in *CIKM*, 2007, pp. 909–914.

[21] M. Barbaro and T. Zeller, "A face is exposed for AOL searcher no. 4417749," *New York Times*, 2006.

[22] G. Karypis and V. Kumar, "Analysis of multilevel graph partitioning," in *SC*, 1995.

[23] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, "Epidemic spreading in real networks: An eigenvalue viewpoint," in *SRDS*, 2003, pp. 25–34.

[24] C. Aggarwal and H. W. (Eds.), *Managing and mining graph data, First edition*. Springer-Verlag, 2010.

[25] E. Zheleva and L. Getoor, "To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles," in *WWW*, 2009, pp. 531–540.

[26] V. Leroy, B. B. Cambazoglu, and F. Bonchi, "Cold start link prediction," in *KDD*, 2010, pp. 393–402.

[27] D. Liben-Nowell and J. M. Kleinberg, "The link prediction problem for social networks," in *CIKM*, 2003, pp. 556–559.