Measuring Anonymity of Pseudonymized Data After Probabilistic Background Attacks

Rajiv Bagai, Nafia Malik, and Murtuza Jadliwala, *Member, IEEE*

***Abstract*— There is clear demand among organizations for sharing their data for mining and other purposes without compromising the privacy of individual objects contained in the data. Pseudonymization is a simple, yet widely employed technique for sanitizing such data prior to its release; it replaces identifying names in the data by pseudonyms. Well-known metrics already exist in the literature for measuring the amount of anonymity still contained in some pseudonymized data in the aftermath of an infeasibility background attack. While the need for a metric for the much wider and more realistic class of probabilistic background attacks has also been well identified, currently no such metric exists. We fulfill that long identified need by presenting two metrics, an approximate and a more exact one, for measuring anonymity in pseudonymized data in the wake of a probabilistic attack. These metrics are rather intractable, thus impractical to employ in real-life situations. Therefore, we also develop an efficient heuristic for our superior metric, and show the remarkable accuracy of our heuristic. Our metrics and heuristic assist a data owner in evaluating the safety level of pseudonymized data against probabilistic attacks before making a decision on its release.**

***Index Terms*— Pseudonymization, degree of anonymity, infeasibility attacks, probabilistic attacks, combinatorial matrix theory, forecasting errors.**

I. Introduction

THE conflict between reaping the benefits of sharing an organization's data to a data miner or the public, and protecting the privacy or confidentiality of sensitive parts of the data, are all too well recognized. Data owners, such as retail stores, hospitals, and other businesses or government agencies, often have an urge or obligation to release data for reasons such as learning about consumer purchasing trends, compliance with transparency regulations, or even just for the public good. Such data usually contains sensitive information, such as salaries or medical conditions, and a straightforward release of data is not appropriate.

A number of sophisticated techniques have been proposed, over the years, for sanitizing such data prior to releasing, with the aim of hiding sensitive correlations contained in it, while still maintaining its underlying characteristics that are of interest to the release recipients. Among the well known techniques are those that achieve k-anonymity of Sweeney [1], l-diversity of Machanavajjhala et al. [2], t-closeness of Li et al. [3], and their numerous variants. By *generalization* and *suppression* of some data values, these methods lessen the visible correlation between identifying attribute values in the data and confidential information corresponding to those values. Another technique, *randomization* of Agrawal and Srikant [4], and its extensions, work by adding sufficiently large noise to the data by perturbing some of its values. Many other approaches exist, and all are well explained in several works, such as in Chen et al. [5] and Fung et al. [6]. Also, Aggarwal and Yu [7] is a collection of several surveys on relevant issues in the area, like k-anonymous data mining, randomization methods, and anonymity measures.

Our focus in this paper is on the relatively simpler technique of *substitution* of identity revealing items in the data, like social security numbers of people, URLs of websites visited, names of products purchased, etc., by fictitious values, even as simple as positive integers. In contrast to the approaches mentioned above, this data sanitization technique does not perturb data, thereby resulting in higher utility of its sanitized version. Often called *pseudonymization* [8], this technique is, in spirit, akin to substitution ciphers, that map original alphabet symbols to new ones, in a one-to-one fashion, and have been well studied and adopted in secure communication. Konheim [9] contains a detailed explanation and analysis of substitution ciphers.

We are *not* advocating here for pseudonymization to be preferred over the more sophisticated techniques mentioned above. Rather, our work is motivated by the following observations:

1) This data sanitization technique is simple to understand and carry out.
2) It complies with most privacy standards, such as those laid out by the U.S. Health Insurance Portability and Accountability Act, and the European Union's Data Protection Directive. Neubauer and Kolb [10] is a methodical evaluation of its compliancy.
3) Pseudonymization-like techniques are already employed in a variety of domains, in which a secret linkage between members of two sets is attempted to be revealed. Some examples are Abouakil [11] for privacy of images in medical healthcare, Beresford and Stajano [12] for location privacy, Giannotti et al. [13] for privacy-preserving mining of association rules from transaction databases, Kerschbaum [14] for collaborative intrusion detection, Danezis and Troncoso [15] for anonymous communications, and Rottondi et al. [16] for

Manuscript received December 16, 2015; revised August 28, 2016 and December 30, 2016; accepted December 31, 2016. Date of publication January 20, 2017; date of current version February 22, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mauro Barni. *(Corresponding author: Rajiv Bagai.)*

R. Bagai and M. Jadliwala are with the Department of Electrical Engineering and Computer Science, Wichita State University, Wichita, KS 67260-0083 USA (e-mail: rajiv.bagai@wichita.edu).

N. Malik is with Panasonic Automotive Systems, Peachtree City, GA 30269 USA.

Digital Object Identifier 10.1109/TIFS.2017.2656458

securing data collected by smart grid meters. Although presented here in the setting of an organization's database, our results are thus applicable to all such domains.

4) High-profile blunders have been committed by organizations relying on it naïvely, without an adequate understanding of its risky interplay with background knowledge. In 2006, AOL released its pseudonymized user search query log for research purposes, which resulted in several instances of privacy breaches, a lawsuit, and the resignation of AOL's Chief Technology Officer [17]. Around the same time, Netflix released a pseudonymized user movie rating data for improving its rating predictions. Major security holes in the data were exposed in Narayanan and Shmatikov [18], leading to another lawsuit [19].

The main aspect of pseudonymization we study in this paper is the safety level it provides, or the anonymity level that remains in any pseudonymized data, in the aftermath of a background attack, specifically, a *probabilistic* attack, as defined below.

If t sensitive data items are pseudonymized using an equal number of pseudonyms, in a one-to-one fashion, then there are $t!$ possible associations (or one-to-one perfect *matchings*) between these objects. The data owner employs exactly one of these matchings to achieve the pseudonymization, and intends to keep that matching hidden from any adversary among all other possible matchings. For large values of t, by itself, this provides an acceptable amount of anonymity. An attacker, often armed with some background knowledge of the domain, is capable of making inroads towards uncovering the all important matching employed by the data owner. Two kinds of background attacks are well known in this context:

- **Infeasibility Attacks:** These attacks determine infeasibility of some of the possible matchings of being the one employed by the data owner.
- **Probabilistic Attacks:** These attacks arrive at a probability for each possible matching of being the one employed by the data owner.

As shown later in the paper, the class of infeasibility attacks is a finite subclass of the uncountably infinite class of probabilistic attacks.

Two metrics exist in the literature for measuring the anonymity remaining in the pseudonymized data upon conclusion of an infeasibility attack. Edman et al. [20] proposed a metric based upon the number of matchings that are still deemed feasible by the attacker, after the attack. The metric of Lakshmanan et al. [21] is an improvement, in that it considers not just the raw number of matchings that still appear feasible to the attacker but, in a sense, the average "correctness" of those seemingly feasible matchings.

While Lakshmanan et al. [21] identified a need for arriving at a metric for measuring anonymity after a probabilistic attack, they left development of such a metric as an important future work.

In this paper, we undertake that task of developing such a metric. As a first step, we present a rough metric, Δ, based on Shannon-entropy [22], and show that this metric is in fact a generalization, for probabilistic attacks, of the metric of Edman et al. [20]. The measurement technique underlying this metric was introduced earlier in our preliminary work [23], for measuring anonymity in communication systems.

Our main contribution in the current paper, however, is a more accurate metric, Ψ, which we show to be a generalization, for probabilistic attacks, of the improved metric of Lakshmanan et al. [21]. Both of our metrics, Δ and Ψ, are unified, in that they work for infeasibility as well as probabilistic attacks, and both are likely not computable accurately in polynomial-time (in terms of the dataset size t), as their values depend upon permanent values of certain real matrices underlying the model, and computation of permanents is known to be #P-complete [24]. We therefore develop an efficient heuristic for Ψ, and show that although our heuristic can be computed in just linear-time, it is a fairly accurate approximation of our metric Ψ.

The rest of this paper is organized as follows. In Section II, we give an overview of the existing methods for measuring anonymity after an infeasibility attack, and a comparison of those approaches. In Section III, we generalize our attack model into one that results in a possibly uneven probability distribution on the set of all matchings. Section IV develops our two metrics, Δ and Ψ, and shows that they are probabilistic generalizations of the existing metrics for infeasibility attacks. In Section V, we present our heuristic and demonstrate it to be an acceptably close approximation of our metric Ψ. Section VI contains a brief comparison of pseudonymization with other well-known privacy techniques, and outlines the overall scope of our metrics. Finally, Section VII concludes our work and gives some directions for future work.

II. Measuring Anonymity After Infeasibility Attacks

In this section we give a brief overview of the methods proposed by Edman et al. [20] and Lakshmanan et al. [21] to determine the safety level of some pseudonymized data, after some partial information about the data somehow gets leaked to an attacker, leading to an infeasibility attack. Their methods give different measures of the level of anonymity remaining in the data in the aftermath of the attack, and help the data owner to answer the all important question of whether or not the pseudonymized data is still safe to be released.

A. The Attack Model

Let X be a nonempty universe of t sensitive data items that need to be pseudonymized, such as social security numbers of people, IP addresses of machines, or web URLs visited. Let Y be a set of pseudonyms for items in X, such that each data item in X corresponds to a unique pseudonym in Y, and *vice versa*. Thus, $|X| = |Y| = t$, and preferably, $X \cap Y = \emptyset$. While the attacker is assumed to know both X and Y, the data owner attempts to keep secret the exact correspondence between elements of X and Y. The maximum anonymity that the owner can strive to achieve is when for any particular item $x \in X$, each member of Y appears to be a feasible candidate for being the pseudonym of x. This situation is depicted by

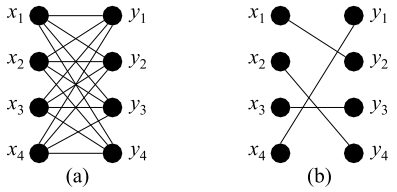


Fig. 1. (a) Complete anonymity; (b) An instance of no anonymity.

the complete bipartite graph $K_{t,t}$ between X and Y, as shown in Fig. 1(a) for $t = 4$. Any edge $\langle x_i, y_j \rangle$ in this graph indicates that the original data item x_i is possibly being pseudonymized by the pseudonym y_j.

The attacker, on the other hand, carries out an attack based on some background knowledge of the application domain, which results in determining the infeasibility of some edges in the complete bipartite graph of Fig. 1(a). These edges are then removed by the attacker from the graph. After a completely successful attack, for each pseudonym $y \in Y$, the attacker would have identified exactly one original item $x \in X$ that could have been pseudonymized by y. In other words, the attacker would have obtained a *perfect matching* (or just *matching*) between X and Y. In this case, the data is thus considered to have been left with no anonymity. There are $t!$ possible matchings between X and Y, an arbitrary one of which is shown in Fig. 1(b).

B. Some Attack Examples

Figs. 1(a) and 1(b) correspond to the two extreme situations, namely complete anonymity and no anonymity at all. In general, after having detected some edges as infeasible, an attack would result in a bipartite graph that lies somewhere between these two extreme ends. Exactly which edges of the complete bipartite graph are missing from the graph resulting from the attack will depend upon how much and what kind of information is available to that attack.

As a simple example, let $X = \{\text{Bread, Eggs, Milk, Sausages}\}$ be the universe of items of a grocery store, and $Y = \{a, b, c, d\}$. Suppose the store releases its database of 10 anonymized transactions, as shown in Fig. 2(a), where each of these transactions is the set of pseudonyms of items purchased in a single order. Now suppose the attacker has background knowledge that Bread is bought in 50% to 80% of all orders. This is shown as the attacker's expected frequency range of this item in Fig. 2(b), along with ranges of all other items in X. As only two of the 10 released transactions contain the pseudonym a, the attacker's observed frequency of a is 20%, also shown in Fig. 2(b), along with the observed frequencies of all other pseudonyms in Y.

From the above, it can be concluded that the item Bread must have been anonymized as either c or d, because the observed frequencies of others, namely a and b, are outside the expected frequency range of Bread. Similar reasoning can be performed on all other items in X to arrive at the reduced graph produced by this attack, also shown in the figure. Note that in this graph edges such as $\langle \text{Bread}, a \rangle$ and $\langle \text{Bread}, b \rangle$ are missing as they were determined by the attacker as infeasible. Fig. 2(c) shows the biadjacency matrix of this graph.

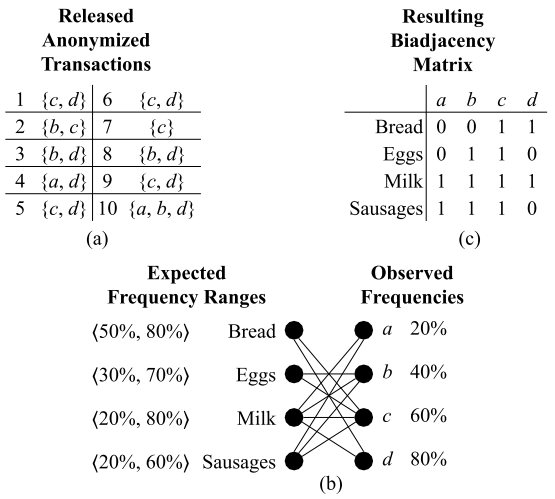


Fig. 2. (a) Anonymized transactions released by owner; (b) Graph resulting from attack by removing edges determined to be infeasible; (c) Biadjacency matrix of this graph.

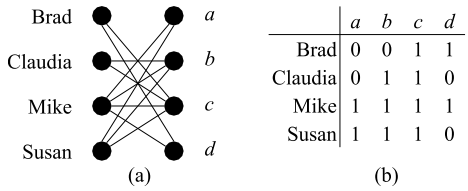


Fig. 3. (a) Graph and (b) biadjacency matrix resulting from an infeasibility attack on the employee database.

As another example, consider the employee database of an organization, containing an identifying attribute *Employee Name*, along with some quasi-identifier attributes, like *Zip Code*, *Age*, *Marital Status*, etc. Suppose this data is released to public, after substituting all employee names in it by unique pseudonyms. An adversary with access to the usually publicly available names of employees on the organization's roll and other public databases, such as the census database or voter registration list, can exploit common quasi-identifier information across datasets to narrow down the feasible employee name and pseudonym combinations. Such an infeasibility attack can result in a bipartite graph and biadjacency matrix, as shown in Fig. 3, which are similar to those in Fig. 2.

All graphs mentioned in the rest of this paper are bipartite. Recall that any matching, an example of which appeared earlier in Fig. 1(b), is essentially a graph, in which each vertex has *exactly* one edge connected to it.

C. Existing Anonymity Metrics, d and $\mathcal{E}$

To decide whether or not to release the pseudonymized data, the data owner needs to assess the risk of disclosure. In this section, we summarize two existing proposals in the literature for helping the data owner make this assessment. The metric d of Section II-C1 was proposed by Edman et al. [20]. The metric $\mathcal{E}$ of Section II-C2, is a more accurate one for the same task, and was constructed by Lakshmanan et al. [21]. It also appeared earlier in their preliminary work [25].

1) The Metric d of Edman et al. [20]: For any matching M and graph G, we say M is *contained* in G, if all edges of M are also in G. It is well known (see, for example, Asratian et al. [26]) that the number of matchings contained in G is the same as the permanent of the 0-1 biadjacency matrix of G. The *permanent* of any $t \times t$ matrix A of real numbers is defined as:

$$\text{per}(A) = \sum\nolimits_{\phi \in \Phi_t} A_{1\phi(1)} A_{2\phi(2)} \cdots A_{t\phi(t)},$$

where Φ_t is the set of all bijections $\phi : \{1, 2, \ldots, t\} \rightarrow \{1, 2, \ldots, t\}$, i.e. permutations of the first t positive integers. It is also known that if A is a 0-1 matrix, then per(A) is an integer between 0 and $t!$. We let $\mathcal{M}(G)$ denote the set of all matchings contained in G.

Let M_O denote the matching employed by the data owner to pseudonymize items in X by pseudonyms in Y. If G is a graph resulting from an attack, then $\mathcal{M}(G)$ is the set of all matchings that still seem feasible to the attacker. Under the assumption that the attacker does not incorrectly label any edge of $K_{t,t}$ as infeasible, it follows that $M_O \in \mathcal{M}(G)$, thus $|\mathcal{M}(G)| \geq 1$.

The size of $\mathcal{M}(G)$, as proposed by Edman et al. [20], is a crude measure of the level of data anonymity remaining after the attack, as that is the extent to which M_O is hidden among all matchings that still seem feasible to the attacker. They proposed the following as the anonymity metric:

$$d(G) = \begin{cases} 0 & \text{if } t = 1, \\ \dfrac{\log(|\mathcal{M}(G)|)}{\log(t!)} & \text{otherwise.} \end{cases}$$

If the value of the above metric is above some acceptable threshold, the sanitized data may be considered *safe* for release by the data owner. The metric $d(G)$ is reasonable as it compares the number of matchings deemed feasible by the attacker with the number of all possible matchings. Note that $0 \leq d(G) \leq 1$. Also, $d(G) = 0$ iff $\mathcal{M}(G)$ has just one matching, i.e. there remains no anonymity, and $d(A) = 1$ iff $t > 1$ and $\mathcal{M}(G)$ has $t!$ matchings, i.e. full anonymity.

The example graph of Fig. 3(a) contains 5 matchings out of the 24 maximum possible. By the above metric, the degree of data anonymity after the attack resulting in that graph is $\log(5) / \log(24) \approx 0.506$.

2) The Metric $\mathcal{E}$ of Lakshmanan et al. [21]: A more accurate anonymity measure can be arrived at by taking into account the collective "correctness" of matchings in $\mathcal{M}(G)$. Let any edge e of an arbitrary matching M, be called a *crack*, if e is also in M_O. The number of cracks in M, denoted by $C(M)$, gives a measure of the "correctness" of M. Clearly, $C(M_O) = t$, and for any other matching M, $0 \leq C(M) < t$. Different matchings in $\mathcal{M}(G)$ have, in general, different numbers of cracks. The expected number of cracks in a randomly chosen matching from $\mathcal{M}(G)$ gives a better measure of the extent to which the attacker has succeeded in breaking anonymity. If this expected value is below some acceptable threshold, the sanitized data may be considered safe for release by the data owner.

Lakshmanan et al. [21] formulated an expression for the expected number of cracks in a randomly chosen matching from $\mathcal{M}(G)$, under the important assumption that all matchings in $\mathcal{M}(G)$ are equally likely. Let A_G be the $t \times t$ biadjacency matrix of G. As stated earlier, $|\mathcal{M}(G)| = \text{per}(A_G)$. We first compute the number of matchings in $\mathcal{M}(G)$ that have exactly c cracks. Let $X^c = \{S \subseteq X : |S| = c\}$ be the set of all subsets of X of size c. For any $S \in X^c$, let $G(S) = \langle X', Y', E' \rangle$ be the graph obtained from $G = \langle X, Y, E \rangle$ after removing some of its vertices and edges as follows:

1) $X' = X \setminus S$, i.e. all vertices in S are removed from X;
2) $Y' = Y \setminus Y''$, where $Y'' = \{y : x \in S \text{ and } \langle x, y \rangle \in M_O\}$, i.e. all such vertices are removed from Y that are counterparts in M_O of vertices in S;
3) $E' = E \setminus E''$, where $E'' = \{\langle x, y \rangle : x \in S \text{ or } y \in Y'' \text{ or } (x \in X' \text{ and } \langle x, y \rangle \in M_O)\}$, i.e. all such edges are removed from E that are either incident to vertices removed from X or Y, or are counterparts in M_O of vertices remaining in X'.

Now, per($A_{G(S)}$) is the number of matchings in $\mathcal{M}(G)$ whose cracks are incident to the c vertices in S. The total number of matchings in $\mathcal{M}(G)$ that have exactly c cracks is therefore:

$$\sum\nolimits_{S \in X^c} \text{per}(A_{G(S)}).$$

The expected number of cracks in a randomly chosen matching from $\mathcal{M}(G)$ is thus given by:

$$\mathcal{E}(G) = \frac{1}{\text{per}(A_G)} \left[\sum\nolimits_{c=0}^{t} \left\{ c \sum\nolimits_{S \in X^c} \text{per}(A_{G(S)}) \right\} \right].$$

As an example, suppose

$$M_O = \{\langle \text{Brad}, d \rangle, \langle \text{Claudia}, b \rangle, \langle \text{Mike}, c \rangle, \langle \text{Susan}, a \rangle\}$$

was the actual matching employed by the owner for pseudonymization of the database upon which the attack of Fig. 3 was conducted. The 5 matchings contained in the graph of Fig. 3(a) can then be seen to have 1, 2, 2, 2, and 4 cracks, respectively. The expected number of cracks in a matching chosen randomly from these is thus $(1+2+2+2+4)/5 = 2.2$.

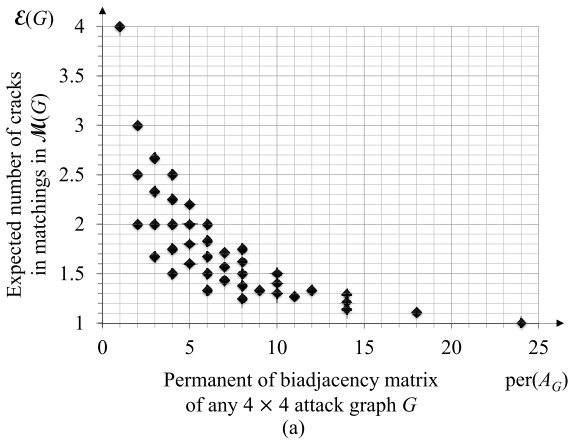
D. An Analysis of Existing Metrics

A closer inspection of the existing anonymity metrics summarized in Section II-C reveals that while these metrics often agree with each other, there are numerous situations in which their results lead to different decisions. This is due to the fact that the metric $d(G)$ depends only on the permanent of the underlying biadjacency matrix, i.e. the number of matchings that appear feasible to the attacker, whereas the metric $\mathcal{E}(G)$ looks at the "correctness" of each of those matchings, thereby resulting in a more accurate measure of anonymity.

To illustrate their different behaviors on our attack example of Fig. 3, let us again assume that

$$M_O = \{\langle \text{Brad}, d \rangle, \langle \text{Claudia}, b \rangle, \langle \text{Mike}, c \rangle, \langle \text{Susan}, a \rangle\}$$

was the matching adopted by the data owner to arrive at the released pseudonymized database. Consider now the set of all 4×4 possible biadjacency matrices corresponding to all attack graphs that contain M_O. Fig. 4(a) plots the permanents of all such matrices, along with the expected crack values of their



(a)

	a	b	c	d
Brad	0	1	0	1
Claudia	1	1	1	0
Mike	1	0	1	1
Susan	1	0	0	1

(b)

	a	b	c	d
Brad	1	1	1	1
Claudia	0	1	1	0
Mike	1	1	1	1
Susan	1	0	1	0

(c)

Fig. 4. (a) Permanents versus expected crack values for all possible 4×4 attack graphs; (b) An example matrix with permanent 4 and expected crack value 1.5; (c) An example matrix with permanent 6 and expected crack value 2.

underlying attack graphs. Although there are $2^{12} = 4096$ distinct such matrices, many of them possess the same permanent and expected crack value, leading to far fewer distinct points on this plot.

From the decreasing trend of the data points in Fig. 4(a), it is evident that, in general, both metrics agree, as the higher the remaining anonymity after an attack, the higher the permanent value, and the lower the expected crack value. However, sometimes these metrics disagree. Observe that, for most permanent values, several different expected crack values exist. For example, several matrices exist, all with the permanent value 4, with expected crack values ranging from 1.5 to 2.5. Thus, while the metric $d(G)$ evaluates the underlying anonymity of all these matrices to be the same, the metric $\mathcal{E}(G)$ results in different anonymity levels for them.

A more interesting phenomenon is depicted by the points $\langle 4, 1.5\rangle$ and $\langle 6, 2\rangle$. Fig. 4(b) shows a representative matrix for $\langle 4, 1.5\rangle$, as its permanent is 4 and the expected crack value can be shown to be $(0 + 1 + 1 + 4)/4 = 1.5$. Similarly, Fig. 4(c) shows a representative matrix for $\langle 6, 2\rangle$. According to the $d(G)$ metric, the former matrix corresponds to lower anonymity, as it hides M_O among just 4 matchings, while the latter hides it among 6 matchings. However, according to the $\mathcal{E}(G)$ metric, the former matrix corresponds to higher anonymity, as its expected crack value of 1.5 is lower than that of the latter matrix, namely 2. Thus, in this case, the two metrics completely disagree with each other.

III. Uneven Probability Distributions on Matchings

An important assumption underlying the metric $\mathcal{E}(G)$ of Lakshmanan et al. [21] is that all matchings in $\mathcal{M}(G)$ seem *equally likely* to the attacker. While they recognized the need for a metric for the general case, where this assumption does not necessarily hold, they left the development of such a metric as beyond the scope of their work.

$\boldsymbol{P}$	a	b	c	d
Brad	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
Claudia	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
Mike	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$
Susan	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

Fig. 5. An example probability matrix, resulting from a probabilistic attack on the employee database of Section II-A.

In this section, we generalize the attack model outlined in Section II for the general scenario in which, due to additional knowledge available to the attacker, the attack results in a possibly uneven probability distribution on the set of all matchings. Later, in Section IV, we will develop metrics for attacks in our new model.

A. Probabilities on Graph Edges

The attacks considered in Section II resulted in rendering some of the edges in the complete graph $K_{t,t}$ between X and Y as infeasible, i.e. determined with full certainty as not being in the matching M_O employed by the data owner to pseudonymize items in X by those in Y. This enabled the attacker to arrive at a subgraph G that has the same vertices as in $K_{t,t}$, but just its feasible edges, usually fewer than the t^2 edges of $K_{t,t}$. The subgraph G was represented by a $t \times t$ 0-1 biadjacency matrix A_G that contains the value 1 for every feasible edge and the value 0 for every edge ruled out by the attacker as infeasible.

Often, the attacker is not able to completely rule out, with absolute certainty, the possibility of some edge of $K_{t,t}$ being in M_O, but is only able to assign some real-valued probability to this event. Such probabilistic attacks too undermine privacy, and have already been recognized in many domains other than data privacy, such as for privacy in social networks data publishing [27], [28], location privacy [12], [29], trajectory privacy [30], [31], and anonymity in persistent communications [15]. These domains do not contain any data pseudonymization per se, but still contain some secret linkage between members of two sets that an attacker usually attempts to reveal. Although our results are applicable to all such domains, here we develop them solely in the context of data pseudonymization.

Unlike an infeasibility attack that results essentially in a 0 or 1 label on each edge of $K_{t,t}$, indicating whether or not it is possible for that edge to be in M_O, a probabilistic attack assigns a real value between 0 and 1 as label to each edge of $K_{t,t}$. The label assigned to an edge is the probability of that edge, according to the attacker, of belonging to M_O. These values can be arranged as a *probability matrix*, an example of which is shown in Fig. 5. In this example attack, on the employee database mentioned in Section II-A, the attacker has completely ruled out the possibility of pseudonym a of representing either Brad or Claudia, but has assigned each of Mike and Susan a probability of $1/2$ of being pseudonymized by that pseudonym. Similarly, the pseudonym b is considered

to be representing Brad and Claudia, each with a probability of 1/3, and Mike and Susan with 1/6.

B. Probability Distribution on Matchings

An important characteristic of a probability matrix produced by such an attack is that it is *doubly-stochastic*, i.e. the sum of all values in any of its rows or columns is 1. This follows from the fact that M_O is essentially a bijection between $X = \{x_1, x_2, \ldots, x_t\}$ and $Y = \{y_1, y_2, \ldots, y_t\}$. We now show that such a matrix induces a probability distribution on the set $\mathcal{M}(K_{t,t})$ of all possible matchings.

Let a *slice* of a $t \times t$ matrix P be any subset of its cells that contains exactly one cell from each row of P. Each slice therefore has exactly t cells. Additionally, a slice of P is a *diagonal* if no two of its cells lie in the same column of P. Let $\mathcal{S}(P)$ and $\mathcal{D}(P)$ denote, respectively, the sets of all slices and diagonals of P. Note that, if P is a probability matrix, a cell in P corresponds to an edge of the system's complete bipartite graph $K_{t,t}$ between X and Y, a slice corresponds to a subgraph of that graph obtained by removing all but one edge connected to each $x \in X$ (i.e. a function from X to Y), and a diagonal corresponds to a matching between X and Y. Clearly, P has t^t slices, of which $t!$ are diagonals.

Let the *weight* of any slice s of P, denoted $\omega(s)$, be the product of values in all cells of s. The following proposition is straightforward.

Proposition 1: For any probability matrix P,

(a) $\sum_{s \in \mathcal{S}(P)} \omega(s) = 1$, and
(b) $\sum_{d \in \mathcal{D}(P)} \omega(d) = \text{per}(P)$.

Proof: (a) Recall that P is $t \times t$. By definitions and algebraic rearrangement we have,

$$\sum\nolimits_{s \in \mathcal{S}(P)} \omega(s) = \sum\nolimits_{j_1=1}^{t} \sum_{j_2=1}^{t} \cdots \sum\nolimits_{j_t=1}^{t} P_{1j_1} P_{2j_2} \cdots P_{tj_t}$$
$$= \prod\nolimits_{i=1}^{t} (P_{i1} + P_{i2} + \cdots + P_{it}) = 1.$$

The last equality follows from the fact that the sum of each row of P is 1.

(b) Follows immediately from the definition of per(P). ■

In other words, per(P) is the sum of weights of all diagonals of P. As $\mathcal{D}(P) \subseteq \mathcal{S}(P)$, a corollary of the above proposition is that per(P) ≤ 1. The equality holds when P contains exactly one 1 in each of its rows and columns. The minimum possible value of per(P) is well known to be $t!/t^t$, when all entries in P are $1/t$ (see, for example, Egorychev [32]).

We let $\mathcal{W}(d) = \omega(d)/\text{per}(P)$ be the *normalized weight* of any diagonal $d \in \mathcal{D}(P)$. The following proposition follows from Proposition 1(b).

Proposition 2: For any probability matrix P,

$$\sum\nolimits_{d \in \mathcal{D}(P)} \mathcal{W}(d) = 1.$$

As the values contained in the matrix P are probabilities, and the sum of values in each row of P is 1, each row is essentially a probability distribution on the set Y. The values contained in any particular row i are the probabilities for each $y_j \in Y$ of being connected with x_i in the matching M_O employed by the data owner.

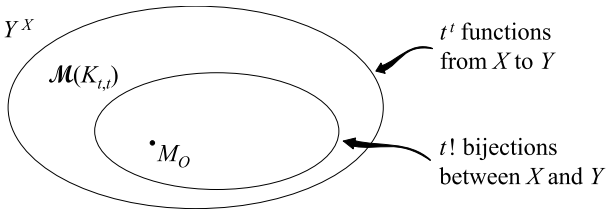


Fig. 6. Sets Y^X of all t^t functions from X to Y, and $\mathcal{M}(K_{t,t})$ of all $t!$ bijections between X and Y.

Consider now the set Y^X, shown in Fig. 6, of all t^t functions $f : X \to Y$, and let some fixed function $g \in Y^X$ be given. Suppose a function f from the set Y^X is constructed randomly as follows:

1) We choose some $y_j \in Y$, with probability P_{1j} according to the distribution contained in the first row of P, and set that chosen y_j to be $f(x_1)$.
2) We similarly set $f(x_2), f(x_3), \ldots, f(x_t)$ according to the distributions contained in rows $2, 3, \ldots, t$, respectively.

The probability that the function f constructed in this fashion is identical to the given function $g \in Y^X$ is $\prod\{P_{ij} \mid g(x_i) = y_j\}$, i.e. the weight of the slice of P that corresponds to g. By Proposition 1(a), these weights add up to 1, i.e. we have a probability distribution on the entire set Y^X. Moreover, by Proposition 1(b), the probability that our randomly constructed function f is a bijection, i.e. it represents a matching between X and Y, is per(P).

Now suppose the given function g is a bijection, i.e. $g \in \mathcal{M}(K_{t,t})$. Then, given the event that the function f constructed randomly as above is also a bijection, the *normalized* weight of the diagonal of P corresponding to g is the probability of the event: $f = g$. This yields a probability distribution on the set $\mathcal{M}(K_{t,t})$ since, by Proposition 2, these normalized weights add up to 1.

As the values in P are the probabilities of edges being in M_O (similar to f above), the normalized weights of the individual diagonals of P (corresponding to all possible bijections g, as above) are thus the probabilities associated by P to their corresponding matchings of being M_O.

In other words, any probability matrix P resulting from an attack, such as the one in Fig. 5, induces a probability distribution over $\mathcal{M}(K_{t,t})$. In this distribution, the probability induced by P on any particular matching $M \in \mathcal{M}(K_{t,t})$ is the probability, arrived at by the attack, of the event $M = M_O$.

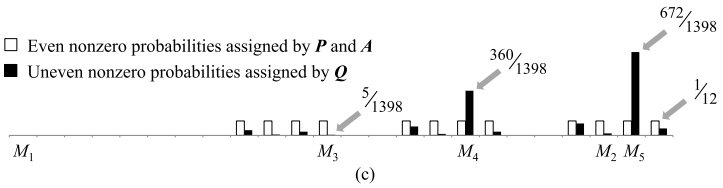
It is worth noting that, by definition of a probability distribution, each such induced probability value is some real number between 0 and 1, and more importantly, the sum of these probability values, over all $M \in \mathcal{M}(K_{t,t})$, is 1. Thus the probability assigned by P to any function in Y^X that is not in $\mathcal{M}(K_{t,t})$ is 0. Our mathematical framework thereby guarantees that two or more different items in X are never associated with the same item in Y, or *vice versa*. As an example, although the matrix P of Fig. 5 explicitly states that the probabilities of the two events $\langle \text{Claudia}, b \rangle \in M_O$ and $\langle \text{Mike}, b \rangle \in M_O$ are 1/3 and 1/6, respectively, the joint probability of these two events, which associate different items

A	a	b	c	d
Brad	0	1	1	1
Claudia	0	1	1	1
Mike	1	1	1	1
Susan	1	1	1	1

(a)

Q	a	b	c	d
Brad	0	$1/8$	$3/8$	$1/2$
Claudia	0	$1/4$	$7/16$	$5/16$
Mike	$3/4$	$1/8$	$1/16$	$1/16$
Susan	$1/4$	$1/2$	$1/8$	$1/8$

(b)



(c)

Fig. 7. (a) The biadjacency matrix A containing same information as the flat matrix P of Fig. 5; (b) An example probability matrix Q that assigns truly uneven probabilities to matchings declared feasible by P and A; (c) Matrices P and A assign even probability of $1/12$ to all 12 feasible matchings, but Q assigns uneven probabilities ranging from $5/1398$ to $672/1398$ to those 12 matchings.

in X with the same item in Y, is guaranteed by our framework to be 0.

C. Flat and Non-Flat Probability Matrices

The permanent of the matrix P of Fig. 5 can be seen to be 1/9. The following are two example matchings, of all the $4! = 24$ matchings contained in the graph $K_{4,4}$ corresponding to this matrix, along with their probabilities of being M_O:

$$M_1 = \{\langle \text{Brad}, a\rangle, \langle \text{Claudia}, b\rangle, \langle \text{Mike}, c\rangle, \langle \text{Susan}, d\rangle\},$$
$$\mathcal{W}(M_1) = \frac{1}{1/9}\left(0 \cdot \frac{1}{3} \cdot \frac{1}{6} \cdot \frac{1}{6}\right) = 0;$$
$$M_2 = \{\langle \text{Brad}, d\rangle, \langle \text{Claudia}, b\rangle, \langle \text{Mike}, c\rangle, \langle \text{Susan}, a\rangle\},$$
$$\mathcal{W}(M_2) = \frac{1}{1/9}\left(\frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{6} \cdot \frac{1}{2}\right) = \frac{1}{12}.$$

A peculiar characteristic of this particular matrix merits a closer look. The normalized weight of 12 of its 24 diagonals is 0, and that of each of the other 12 is $1/12$. Matchings M_1 and M_2 given above belong to those classes, respectively. In other words, matchings that have a non-zero probability of being M_O are all *equally likely* to be M_O.

Probability matrices with this property are called *flat* matrices, and they provide no additional probabilistic information to the attacker than their corresponding 0-1 biadjacency matrices that possess an identical zero-pattern. The biadjacency matrix A of Fig. 7(a) has the same zero-pattern as that of the above probability matrix P, i.e. both matrices contain 0 values in exactly the same cells, thereby declaring the same 12 matchings to be feasible, and all those matchings have, according to P, an equal probability of $1/12$ of being M_O. This is depicted in Fig. 7(c).

On the other hand, the probability matrix Q of Fig. 7(b) possesses the same zero-pattern, but is not flat. The permanent of Q is $699/4096$. According to Q, the probabilities of those 12 feasible matchings of being M_O vary from $5/1398 \approx 0.004$ to $336/699 \approx 0.481$, as shown by the following three example matchings:

$$M_3 = \{\langle \text{Brad}, b\rangle, \langle \text{Claudia}, c\rangle, \langle \text{Mike}, d\rangle, \langle \text{Susan}, a\rangle\},$$
$$\mathcal{W}(M_3) = \frac{1}{699/4096}\left(\frac{1}{8} \cdot \frac{5}{16} \cdot \frac{1}{16} \cdot \frac{1}{4}\right) = \frac{5}{1398} \approx 0.004;$$
$$M_4 = \{\langle \text{Brad}, c\rangle, \langle \text{Claudia}, d\rangle, \langle \text{Mike}, a\rangle, \langle \text{Susan}, b\rangle\},$$
$$\mathcal{W}(M_4) = \frac{1}{699/4096}\left(\frac{3}{8} \cdot \frac{5}{16} \cdot \frac{3}{4} \cdot \frac{1}{2}\right) = \frac{360}{1398} \approx 0.258;$$
$$M_5 = \{\langle \text{Brad}, d\rangle, \langle \text{Claudia}, c\rangle, \langle \text{Mike}, a\rangle, \langle \text{Susan}, b\rangle\},$$
$$\mathcal{W}(M_5) = \frac{1}{699/4096}\left(\frac{1}{2} \cdot \frac{7}{16} \cdot \frac{3}{4} \cdot \frac{1}{2}\right) = \frac{672}{1398} \approx 0.481.$$

As shown in Fig. 7(c), the matrix Q assigns truly uneven probabilities to exactly those matchings that are declared feasible, and are assigned even probabilities, by A and P.

For any graph G, let $\mathfrak{P}(G)$ be the set of all (doubly-stochastic) probability matrices that assign non-zero probabilities to exactly those matchings that are in $\mathcal{M}(G)$. It is easily seen that as long as $\mathcal{M}(G)$ contains at least two matchings, $\mathfrak{P}(G)$ is uncountably infinite. Still, exactly one of the matrices in $\mathfrak{P}(G)$ is flat, a fact that follows from Corollary 2.6.6 in Bapat and Raghavan [33], and all other matrices in $\mathfrak{P}(G)$ assign truly uneven probabilities to matchings in $\mathcal{M}(G)$.

D. Construction of Flat Matrix From Biadjacency Matrix

An interesting aside deserves a brief mention here. Bagai et al. [23] outlined a method based on matrix scalings for arriving at this flat matrix in $\mathfrak{P}(G)$ from the graph's biadjacency matrix A_G. Figs. 8(a) and 8(b) show the biadjacency matrix A_G and the flat matrix $F \in \mathfrak{P}(G)$, respectively, for an example graph G. Note that the zero-pattern of A_G is contained in that of F. In particular, while the edge $\langle \text{Mike}, d\rangle$ is feasible according to A_G, F assigns a zero probability to that edge. This is due to the fact that that edge does not appear in any matching in $\mathcal{M}(G)$, and is thus superfluous.

Another elegant characterization of $F \in \mathfrak{P}(G)$, as the limit of an infinite sequence of matrices, appeared much earlier in Sinkhorn and Knopp [34]. Let f, g and h be functions from and to $t \times t$ real matrices, defined as follows:

$$f(M)_{ij} = M_{ij} \;/\; \sum_{k=1}^{t} M_{ik} \quad \text{(row normalization)}$$
$$g(M)_{ij} = M_{ij} \;/\; \sum_{k=1}^{t} M_{kj} \quad \text{(column normalization)}$$
$$h(M) = g(f(M))$$

Then, $F = \lim_{k\to\infty} h^k(A_G)$. In other words, a procedure that alternately normalizes all rows followed by all columns of A_G, ad infinitum, would converge to F. However, as A_G contains just 0-1 values, intermediate matrices obtained after any finite number of iterations contain only rational values. As the example in Fig. 8 shows, the final solution can be irrational, the limit of an infinite sequence of rational approximations. So in general, this procedure requires an infinite number of iterations. A number of efficient algorithms have therefore been considered, as in Kalantari and Khachiyan [35] and Linial et al. [36], for producing in a finite number of steps, approximate solutions that are within acceptable error bounds.

A_G	a	b	c	d
Brad	0	0	0	1
Claudia	0	1	1	0
Mike	1	0	1	1
Susan	1	1	1	0

(a)

F	a	b	c	d
Brad	0	0	0	1
Claudia	0	$(\sqrt{5}-1)/2$	$(3-\sqrt{5})/2$	0
Mike	$(\sqrt{5}-1)/2$	0	$(3-\sqrt{5})/2$	0
Susan	$(3-\sqrt{5})/2$	$(3-\sqrt{5})/2$	$\sqrt{5}-2$	0

(b)

Fig. 8. (a) The biadjacency matrix A_G of an example graph G; (b) The flat matrix $F \in \mathfrak{P}(G)$.

IV. Metrics for Probabilistic Attacks

In this section, we develop two metrics for measuring anonymity in the wake of an attack that results in an arbitrary probability distribution on the set $\mathcal{M}(K_{t,t})$ of all matchings. These metrics are, respectively, generalizations for our probabilistic attack model, of the metrics of Edman et al. [20] and Lakshmanan et al. [21].

A. A Rough Metric, Δ

As shown in Section III-B, the normalized weights of the diagonals of a probability matrix P are the probabilities associated by P to their corresponding matchings of being M_O. The uncertainty contained in this probability distribution induced by P on the set $\mathcal{D}(P)$ of all its diagonals (or, alternatively, all matchings between X and Y) is a reasonable measure of the anonymity remaining in the system in the aftermath of an attack resulting in P. Ever since the works of Serjantov and Danezis [37] and Diaz et al. [38], Shannon-entropy [22] of such a probability distribution is a well accepted measure of remaining anonymity. In a preliminary work, Bagai et al. [23] employed that technique to arrive at a metric in the context of anonymity in communication systems. Here we show that the metric of Bagai et al. [23] is in fact a generalization of the metric of Edman et al. [20] for our probabilistic attack model.

Let P be any $t \times t$ biadjacency or probability matrix resulting from an attack. We define the underlying system's *degree of anonymity* after this attack as:

$$\Delta(P) = \begin{cases} 0 & \text{if } t = 1, \\ \dfrac{-\sum_{d \in \mathcal{D}(P)} \mathcal{W}(d) \cdot \log(\mathcal{W}(d))}{\log(t!)} & \text{otherwise.} \end{cases}$$

In the above summation, a subexpression $0 \cdot \log(0)$ is interpreted as 0.

Observe that the above metric Δ is a unified metric, as it is for biadjacency *as well as* probability matrices, whereas the metric d of Edman et al. [20], given in Section II-C1, was essentially only for biadjacency matrices as one such is unique for every graph. We first establish that Δ is a generalization of d, by showing that for biadjacency matrices, both of these metrics coincide.

Theorem 1: Let G be any graph with biadjacency matrix A_G, and let $F \in \mathfrak{P}(G)$ be flat. Then, $d(G) = \Delta(A_G) = \Delta(F)$.

Proof: To show the first equality, we recall that the normalized weight of exactly $\text{per}(A_G)$ diagonals of A_G is $1/\text{per}(A_G)$, and that of its remaining diagonals is 0. The numerator of $\Delta(A_G)$ thus becomes:

$$-\text{per}(A_G)\left[\frac{1}{\text{per}(A_G)} \cdot \log\left(\frac{1}{\text{per}(A_G)}\right)\right],$$

which is $\log(\text{per}(A_G)) = \log(|\mathcal{M}(G)|)$, the numerator of $d(G)$ given in Section II-C1.

The second equality follows from the fact that the normalized weight of any diagonal of A_G is the same as that of the corresponding diagonal of F. ∎

Clearly, adding probabilistic information to the edges of any graph should result in a strengthening of the attack, thereby lowering the resulting anonymity. The following result shows that Δ possesses this intuitive property.

Theorem 2: Let G be any graph, and let $P \in \mathfrak{P}(G)$ be not flat. Then, $\Delta(P) < \Delta(A_G)$.

Proof: As $P \in \mathfrak{P}(G)$, the normalized weight of any of its diagonals is 0 iff that of the corresponding diagonal of A_G is 0. Let $\text{per}(A_G) = k > 0$. Then, the normalized weights of exactly k diagonals of A_G is $1/k$ (and that of its remaining diagonals is 0). Let $w_1, w_2, \ldots, w_k$ be the normalized weights of corresponding diagonals of P. As these are the only diagonals of P that may have nonzero weights, their sum is 1. We need to show that:

$$-\sum_{i=1}^{k} w_i \cdot \log(w_i) < -\sum_{i=1}^{k} (1/k) \cdot \log(1/k).$$

Although this property of Shannon-entropy is well known in information theory (see, for example, Kapur [39] for a proof based on Jensen's inequality), here we give a shorter proof.

We first establish the above inequality when all logarithms are natural, i.e. to the base e that has several equivalent characterizations, such as $\sum_{n=0}^{\infty} 1/n!$ and $\lim_{n\to\infty}(1+1/n)^n$, and then generalize the inequality for logarithms to any arbitrary base. It is easily seen that, for all x, $ex \le e^x$, with equality iff $x = 1$. Taking natural logarithms, we have that $1 + \ln(x) \le x$. By substituting $x = (1/k)/w_i$, and simplifying, we get that for all i, $w_i - w_i \cdot \ln(w_i) \le (1/k) - w_i \cdot \ln(1/k)$, with equality iff $w_i = 1/k$. Summation over all i gives:

$$-\sum_{i=1}^{k} w_i \cdot \ln(w_i) \le \ln(k) = -\sum_{i=1}^{k} (1/k) \cdot \ln(1/k).$$

As P is not flat, and normalized weights of at least one pair of corresponding diagonals of any distinct doubly-stochastic matrices must be different from each other, we have that for some i, $w_i \neq 1/k$, leading to a strict inequality.

To generalize the inequality for logarithms to any arbitrary base $b > 0$, since $\ln(x)/\ln(b) = \log_b(x)$, for all x, we simply divide both sides of the inequality by $\ln(b)$. ∎

As an example, consider the graph G whose biadjacency matrix is the matrix A of Fig. 7(a). We already saw that the matrices P of Fig. 5 and Q of Fig. 7(b) are both in $\mathfrak{P}(G)$. P is flat, but Q is not. It is easily verified that $d(G) = \Delta(A) = \Delta(P) = \log(12) / \log(24) \approx 0.782$. The nondecreasing vector

of the 12 nonzero normalized weights of diagonals of Q is:

$$\frac{1}{1398}\langle 5, 7, 12, 16, 30, 30, 42, 56, 72, 96, 360, 672\rangle,$$

resulting in $\Delta(Q) \approx 0.496$. The higher the unevenness in the probability distribution on the matchings contained in G, the lower the anonymity measure given by Δ.

B. An Accurate Metric, Ψ

Just as, by taking M_O into account, the metric $\mathcal{E}(G)$ of Lakshmanan et al. [21] gives a more accurate measure of anonymity than the metric $d(G)$ of Edman et al. [20], we now develop a metric for probabilistic attacks that is more accurate than the $\Delta(P)$ metric given in the previous section.

We first note that, given a graph G, the expected crack value $\mathcal{E}(G)$, of Section II-C2, is essentially the average number of cracks among the matchings contained in $\mathcal{M}(G)$, i.e.

$$\mathcal{E}(G) = \frac{\sum_{M\in\mathcal{M}(G)} C(M)}{|\mathcal{M}(G)|}.$$

For probabilistic attacks, the expected crack value is the *weighted* average number of cracks among all matchings in $\mathcal{M}(K_{t,t})$, where the weight of any matching is its probability of being M_O. As shown earlier, the normalized weights of the diagonals of any biadjacency or probability matrix are the probabilities of their corresponding matchings of being M_O, leading to the following metric.

Let P be any $t\times t$ biadjacency or probability matrix resulting from an attack. The *expected crack value* among all matchings in $\mathcal{M}(K_{t,t})$ after this attack is:

$$\begin{aligned}\Psi(P) &= \frac{\sum_{d\in\mathcal{D}(P)} \mathcal{W}(d)\cdot C(m(d))}{\sum_{d\in\mathcal{D}(P)} \mathcal{W}(d)}\\ &= \sum_{d\in\mathcal{D}(P)} \mathcal{W}(d)\cdot C(m(d)),\end{aligned}$$

where $m(d)$ is the matching in $\mathcal{M}(K_{t,t})$ that corresponds to the diagonal d of P.

Like Δ, the above metric Ψ is also unified, as it is for biadjacency *as well as* probability matrices, whereas the metric $\mathcal{E}$ of Lakshmanan et al. [21] was essentially only for biadjacency matrices as one such is unique for every graph. The following result is similar in nature to Theorem 1 for Δ, as it shows that Ψ is a generalization of $\mathcal{E}$, since for biadjacency matrices, both of these metrics coincide.

Theorem 3: Let G be any graph with biadjacency matrix A_G, and let $F \in \mathfrak{P}(G)$ be flat. Then, $\mathcal{E}(G) = \Psi(A_G) = \Psi(F)$.

Proof: The first equality follows from the fact that for any $d \in \mathcal{D}(A_G)$, $\mathcal{W}(d) = 1/\text{per}(A_G) = 1/|\mathcal{M}(G)|$, if $m(d) \in \mathcal{M}(G)$, and 0 otherwise. The second follows from the already noted fact that the normalized weight of any diagonal of A_G is the same as that of the corresponding diagonal of F. ■

By depending on M_O, the metric Ψ makes a noteworthy departure from Δ, in that adding probabilistic information to the edges of a graph does not always strengthen the underlying attack. In other words, while the property given by Theorem 2 was intuitive for Δ, a similar property is not possessed by Ψ. As an example, let G be the graph whose biadjacency matrix is the matrix A of Fig. 7(a). Recall that the matrices P of Fig. 5 and Q of Fig. 7(b) are both in $\mathfrak{P}(G)$, where P is flat, but Q is not. It can be seen that $\mathcal{E}(G) = \Psi(A) = \Psi(P) = 4/3 \approx 1.333$, as stated by Theorem 3 above. However, $\Psi(Q) = 1183/1398 \approx 0.846 < \mathcal{E}(G)$, i.e. despite assigning extra probabilistic information to the edges of G, Q represents a weaker attack than A. This phenomenon is due to the fact that Q assigns low probabilities to the edges in M_O. While Δ depends on just the unevenness of the probabilities assigned by an attack to each matching for being M_O, Ψ is sensitive, in a sense, to the "correctness" of those probabilities, by taking crack values of matchings into account, thus resulting in a more accurate anonymity measure.

C. Complexity of Computing Δ and Ψ

Unfortunately, both metrics, Δ and Ψ, are hard to compute. Arriving at the normalized weights, $\mathcal{W}(d)$, of the diagonals $d \in \mathcal{D}(P)$ requires computing the permanent of P. Valiant [24] showed that computing the permanent, even if all values of the underlying matrix are just 0 or 1, is #P-complete. A polynomial-time solution for computing the permanent is thus unlikely, as that would imply P = NP. The fastest known exact algorithms for computing the permanent of a $t \times t$ real matrix have time complexity $\Theta(t2^t)$. An example is the method of Ryser that appeared on Page 122 of Minc [40]. Much attention has consequently been given to arriving at approximations to permanents more efficiently, as in Jerrum and Vazirani [41] and Chien et al. [42]. However, even the state-of-the-art polynomial-time approximation method of Jerrum et al. [43] runs in $\mathcal{O}(t^{22})$, which still renders even approximating our metrics, Δ and Ψ, rather infeasible for real-life situations.

It should be noted that as the existing metrics, d and $\mathcal{E}$, summarized in Sections II-C1 and II-C2, respectively, are also based on permanents of matrices, they are, for the very same reason, equally hard to compute, despite their limited capability of measuring anonymity after any of only the comparatively small and finite subclass of infeasibility attacks. In the next section, we present a simple, yet efficient and fairly accurate, heuristic for approximating Ψ.

V. An Efficient Heuristic for Ψ

Let for each i, m_i be the index of the unique item in Y that was actually employed by the data owner to pseudonymize the data item $x_i \in X$, i.e.

$$M_O = \{\langle x_1, y_{m_1}\rangle, \langle x_2, y_{m_2}\rangle, \ldots, \langle x_t, y_{m_t}\rangle\}.$$

Recall that as the sum of values in each row of a given probability matrix P is 1, each of its rows by itself is a probability distribution on the set Y. The values $\langle P_{i1}, P_{i2}, \ldots, P_{it}\rangle$ making up any row i are the probabilities for each $y_j \in Y$ of being connected with $x_i \in X$ in the matching M_O employed by the data owner.

Now, P_{im_i} is the probability with which the attacker correctly associates x_i with y_{m_i}, i.e. the probability with which the attacker "cracks" x_i. This probability is a reasonable estimate

of the "contribution" of index i to the overall expected crack value. $\Psi(P)$, the expected crack value among all matchings in $\mathcal{M}(K_{t,t})$, can thus be estimated by summing this probability across all i, leading to the following heuristic:

$$\mathcal{H}(P) = \sum_{i=1}^{t} P_{im_i}.$$

This heuristic can be computed in $\mathcal{O}(t)$ steps, a vast improvement over $\Psi(P)$, exact computation of which can likely not be done even in polynomial-time.

A. An Alternative Characterization of $\mathcal{H}$

Later, we show that $\mathcal{H}(P)$ is a rather close approximation of $\Psi(P)$, but for now, we observe that they are not identical. $\mathcal{H}(P)$ is, in fact, the expected crack value among all functions in Y^X, which as shown in Fig. 6, is usually a proper superset of $\mathcal{M}(K_{t,t})$. To establish this, we first give the following stronger lemma.

Lemma 1: Let for each $k \in \{1, 2, \ldots, t\}$, $\mathcal{F}_k$ denote the set of all t^k partial functions $f : \{x_1, x_2, \ldots, x_k\} \rightarrow Y$. Then, given M_O and any probability matrix P, we have that for all k, the expected crack value among all partial functions in $\mathcal{F}_k$ is $\sum_{i=1}^{k} P_{im_i}$.

Proof: (By weak induction on k) For the base case, let $k = 1$. $\mathcal{F}_1$ contains t partial functions, exactly one of which has one crack, as it maps x_1 to y_{m_1}. The probability of choosing this function is P_{1m_1}. None of the other $(t-1)$ partial functions in $\mathcal{F}_1$ has any cracks, and the combined probability of choosing one of those is $(1 - P_{1m_1})$. The expected crack value among all partial functions in $\mathcal{F}_1$ is thus $1 \cdot P_{1m_1} + 0 \cdot (1 - P_{1m_1}) = P_{1m_1} = \sum_{i=1}^{k} P_{im_i}$.

For the inductive case, let $2 \leq k \leq t$ and, as the inductive hypothesis, suppose the expected crack value among all of the t^{k-1} partial functions in $\mathcal{F}_{k-1}$ is $\sum_{i=1}^{k-1} P_{im_i}$. $\mathcal{F}_k$ has t^k partial functions, exactly t^{k-1} of which, with a combined probability of P_{km_k}, map x_k to y_{m_k}, thereby adding one crack to each individual partial function in $\mathcal{F}_{k-1}$. The expected crack value among these members of $\mathcal{F}_k$ is thus $1 + \sum_{i=1}^{k-1} P_{im_i}$. The remaining $(t^k - t^{k-1})$ partial functions in $\mathcal{F}_k$, with a combined probability of $(1 - P_{km_k})$, map x_k to other members of Y. As x_k contributes no crack in these partial functions, their expected crack value is still $\sum_{i=1}^{k-1} P_{im_i}$. The expected crack value among all partial functions in $\mathcal{F}_k$ is thus

$$\left(1 + \sum_{i=1}^{k-1} P_{im_i}\right) \cdot P_{km_k} + \left(\sum_{i=1}^{k-1} P_{im_i}\right) \cdot (1 - P_{km_k}),$$

which simplifies to $\sum_{i=1}^{k} P_{im_i}$. ∎

The following theorem follows immediately from the case $k = t$ of the above lemma. The function C in it is extended from matchings to functions in Y^X in a straightforward way.

Theorem 4: Given M_O and any probability matrix P, $\mathcal{H}(P)$ is the expected crack value over all functions in Y^X, i.e. $\mathcal{H}(P) = \sum_{s \in \mathcal{S}(P)} \omega(s) \cdot C(f(s))$, where $f(s)$ is the function in Y^X that corresponds to the slice s of P.

It is interesting to observe that while the expected crack value over Y^X can be computed in just linear-time, that over $\mathcal{M}(K_{t,t})$ can perhaps not be computed in even polynomial-time. We now show that, fortunately, these values are not too far apart. $\mathcal{H}(P)$ can thus be employed as a reasonable heuristic for $\Psi(P)$.

B. Accuracy of the Heuristic $\mathcal{H}$

Clearly, the number of cracks contained in any function $f \in Y^X$, namely $C(f)$, is closely tied to the choice of $M_O \in \mathcal{M}(K_{t,t})$ made by the data owner to pseudonymize items in X by those in Y. There are $t!$ candidates for M_O available to the data owner and, depending upon the choice made, $C(f)$ may range from 0 to t. We extend our notation to reflect this choice. Let μ be any matching in $\mathcal{M}(K_{t,t})$. We then let $C_\mu(f)$ denote the number of cracks in f, given that $M_O = \mu$. In other words, $C_\mu(f)$ is the number of edges common to μ and f. The following proposition will soon be relevant:

Proposition 3: For any function $f \in Y^X$, $\sum_{\mu \in \mathcal{M}(K_{t,t})} C_\mu(f) = t!$.

Proof: Any edge $\langle x, f(x) \rangle$ of $K_{t,t}$ is contained in exactly $1/|Y| = 1/t$ of all the $t!$ matchings in the set $\mathcal{M}(K_{t,t})$. Thus, $\sum_{\mu \in \mathcal{M}(K_{t,t})} C_\mu(f) = \sum_{x \in X} (t!/t) = |X|(t!/t) = t!$. ∎

For a given probability matrix P, the values $\Psi(P)$ and $\mathcal{H}(P)$ therefore also depend upon the choice of M_O, and we extend these notations to reflect that. For any $\mu \in \mathcal{M}(K_{t,t})$, we let $\Psi_\mu(P)$ be the expected number of cracks, under the probabilities contained in P, among all matchings in $\mathcal{M}(K_{t,t})$, given that $M_O = \mu$. We define $\mathcal{H}_\mu(P)$ analogously as the expected number of cracks over all functions in Y^X.

For the 4×4 matrices Q and F of Figs. 7(b) and 8(b), respectively, Fig. 9 plots Ψ_μ and $\mathcal{H}_\mu$, across all possible $4! = 24$ matchings μ. For ease of visualization, the 24 matchings μ are arranged in each plot in an order that results in non-decreasing values of Ψ_μ. These plots provide an informal appreciation of the admirable accuracy with which our heuristic approximates the exact metric, despite the fact that each heuristic value can be computed in just $\Theta(t)$ time, whereas each exact metric value, as mentioned in Section IV-C, takes $\Theta(t2^t)$ time, according to the best algorithm currently known.

For low values of the metric Ψ_μ, the heuristic $\mathcal{H}_\mu$ can be seen in the above plots to be usually overestimating the metric, while for high values of the metric, the heuristic usually underestimates the metric. This phenomenon is typical over all values of t and all $t \times t$ probability matrices.

A rather interesting observation is that across all possible choices of M_O, the total amount of overestimates and underestimates cancel each other out or, loosely speaking, the areas under the curves of our heuristic and the exact metric, over all matchings μ, are identical.

Theorem 5: For any $t \times t$ probability matrix P,

$$\sum_{\mu \in \mathcal{M}(K_{t,t})} \mathcal{H}_\mu(P) = t! = \sum_{\mu \in \mathcal{M}(K_{t,t})} \Psi_\mu(P).$$

Proof: The leftmost term, by Theorem 4, is $\sum_{\mu \in \mathcal{M}(K_{t,t})} \left(\sum_{s \in \mathcal{S}(P)} \omega(s) \cdot C_\mu(f(s))\right)$, which by distributive law becomes $\sum_{s \in \mathcal{S}(P)} \left(\omega(s) \cdot \sum_{\mu \in \mathcal{M}(K_{t,t})} C_\mu(f(s))\right)$. Proposition 3 simplifies this to $t! \sum_{s \in \mathcal{S}(P)} \omega(s)$, which is further reduced, by Proposition 1(a), to $t!$. The reduction of the rightmost term is similar. By definition, it is $\sum_{\mu \in \mathcal{M}(K_{t,t})} \left(\sum_{d \in \mathcal{D}(P)} \mathcal{W}(d) \cdot C_\mu(m(d))\right)$, which can be rearranged

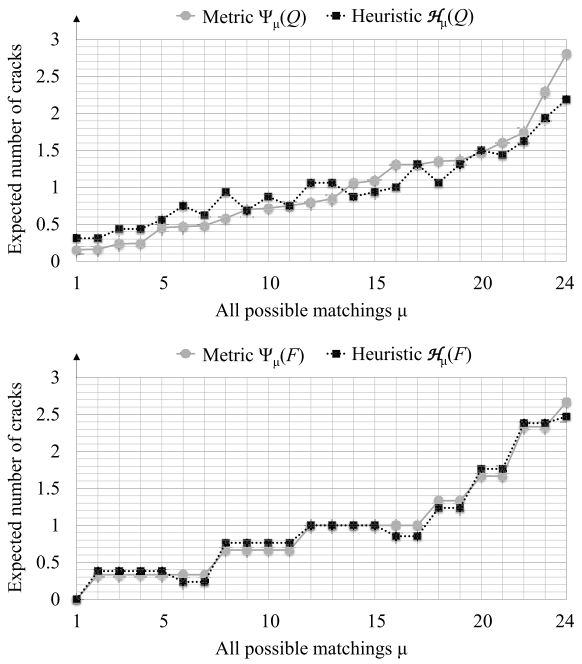


Fig. 9. For the 4×4 matrices Q and F of Fig. 7(b) and Fig. 8(b), respectively, the values Ψ_μ and $\mathcal{H}_\mu$, across all possible $4! = 24$ matchings μ.

to $\sum_{d\in\mathcal{D}(P)} \left(\mathcal{W}(d)\cdot\sum_{\mu\in\mathcal{M}(K_{t,t})} C_\mu(m(d))\right)$. Again, Proposition 3 simplifies this to $t!\sum_{d\in\mathcal{D}(P)}\mathcal{W}(d)$, which by definition of $\mathcal{W}$ is $(t!/\text{per}(P))\sum_{d\in\mathcal{D}(P)}\omega(d)$. Finally, Proposition 1(b), reduces this to $t!$. ■

The above theorem establishes only that the sum total of overestimates made by the heuristic, across all matchings μ, always coincides with the sum total of underestimates. While that is reassuring, the *accuracy* of the heuristic depends entirely on the *extent* to which the heuristic values, for each μ, deviate from the corresponding metric values. Such a deviation is traditionally called *error*. Much work has been done, mainly in the area of time-series forecasting, on techniques for measuring the accuracy of forecasts, such as our heuristic (see Armstrong [44] for basic principles). Several methods exist in the literature for such tasks, like the Root Mean Square Error, Mean Absolute Percentage Error, Geometric Mean Absolute Relative Error, to name a few, each with its own strengths and limitations. The choice of a method, for any particular application, usually depends upon the nature of the underlying data values. Hyndman and Koehler [45], and Shcherbakov et al. [46] are some critical surveys on existing methods.

For our purpose, we chose the *Normalized Mean Absolute Percentage Error* (NMAPE), which is already a method of choice in many other domains, e.g. the wind power forecasting application of Chen et al. [47]. For a given $t\times t$ probability matrix P, this value is given by:

$$\text{NMAPE}(P) = \frac{1}{t!}\times\left(\sum_{\mu\in\mathcal{M}(K_{t,t})}\frac{|\mathcal{H}_\mu(P)-\Psi_\mu(P)|}{t}\right)\times 100.$$

The above employs absolute error values in order to prevent positive and negative error values from canceling each other out. As our heuristic as well as the actual metric values always lie between 0 and t, the largest absolute error is t, which is employed as the normalizing factor to standardize to the scale of 0 to 1. The summation over all μ and division by $t!$ result in averaging these values and, finally, multiplication by 100 expresses this average absolute error on an intuitive percentage scale. NMAPE(P) thus gives the average-case absolute error for P as a percentage of the worst-case. Its values close to 0% indicate accurate computation by the heuristic, while those close to 100% indicate inaccurate computation.

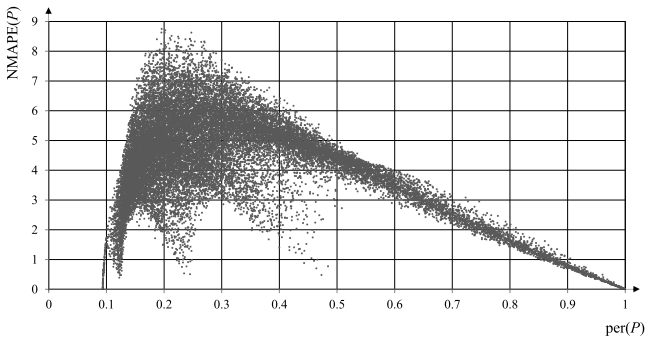


Fig. 10. NMAPE(P) versus per(P), for 24,000 randomly generated 4×4 probability matrices P.

The permanent of a $t\times t$ probability matrix P is a real value in the range $[t!/t^t, 1]$ and, it can be easily shown that in the extreme cases, i.e. when per(P) is either $t!/t^t$ or 1, NMAPE(P) is 0%. To get a better picture of the distribution of NMAPE(P) values over the uncountably infinite space of all probability matrices, we randomly generated 24,000 such matrices of size 4×4. Fig. 10 plots the NMAPE(P) versus per(P) values of these random matrices, and shows that NMAPE(P) is always within 9%, and usually just within 6%, indicating a fairly high degree of heuristic accuracy. Higher values of t exhibit a similar degree of accuracy.

Another noteworthy observation from Fig. 10 is that higher values of NMAPE(P) tend to occur only for relatively lower values of per(P). These are situations where the values in P are more uniform, i.e. the underlying attack is weak. Thus, when our heuristic does err, it errs only on the safer side by perhaps preventing the data owner from releasing data that is in fact secure enough for release. Such a conservative nature makes the heuristic a welcome ally of the data owner.

We end this section with an interesting, far-reaching consequence of Theorem 5.

Corollary 1: A privacy breach in any pseudonymized dataset is the norm.

Proof: $|\mathcal{M}(K_{t,t})| = t!$ and, by Theorem 5, we have that for any $t\times t$ probability matrix P, $\sum_{\mu\in\mathcal{M}(K_{t,t})}\Psi_\mu(P)$ is also $t!$. So the expected value of $\Psi_\mu(P)$, if μ and P are drawn randomly from their respective domains, each with uniform distribution, is $t!/t! = 1$, i.e. one crack is the norm. ■

In other words, even if an attack is picked randomly and not arrived at by any particularly clever analysis by the attacker, one privacy breach should be expected. And more breaches should be expected if the attack is carefully constructed. Though never perfect, for large values of t, the pseudonymization technique may thus be considered acceptable. In any case, the privacy breaches that occurred in the famous AOL and

Netflix cases mentioned in Section I, leading to lawsuits and even resignation, were in fact not surprising.

VI. Comparison With Other Privacy Techniques

In this section, we briefly compare pseudonymization with other well-known privacy techniques, and elucidate the scope of our metrics and heuristic.

A. Differential Privacy

One of the most prominent privacy models studied in recent years is that of differential privacy, proposed initially by Dwork [48]. On some underlying database, let $\mathcal{K}$ be a statistical query answering mechanism that, in the interest of preserving privacy of individuals in the database, produces query answers after adding some randomly-generated noise to exact answers. The mechanism $\mathcal{K}$ is called ϵ-*differentially private* if for any database instances D and D' that differ on at most one record,

$$\Pr[\mathcal{K}(D) \in S] \leq \exp(\epsilon) \cdot \Pr[\mathcal{K}(D') \in S],$$

where S is any set of possible output values of $\mathcal{K}$. This property of the mechanism aims to guarantee each individual that answers produced for the query will be almost identical, whether or not that individual participates in the database, such as in an anonymous survey.

Although surface similarities exist between our work and differential privacy, in that both deal with privacy and probabilities, there are fundamental differences between their underlying scenarios. In pseudonymization, the goal is to release the sanitized data, regardless of the kinds of analyses performed or queries executed on that data later. Privacy provided by the sanitized data is then measured on an attack-by-attack basis. Differential privacy providing methods can be classified into one of two broad categories: interactive and non-interactive (see Dankar and El Emam [49], and Dwork et al. [50]). In interactive methods, no version of the data is ever released. Rather, the query needs to be known in advance, and the extent to which the query is sensitive is measured and used to configure the mechanism's noise level, in order to adequately meet the requirements of the privacy parameter ϵ. The privacy level here is associated with the *computational* mechanism and, once an appropriate mechanism is created, all that is released thereon are sanitized statistical answers produced by it for just that query. In non-interactive methods, although a differentially private sanitized or synthetic data is released, the utility of such data is of an acceptable level for only a small predefined set of queries, with low sensitivities. Releasing data that is of utility for random queries is still a challenge, as is the efficiency of the query answering mechanism.

The parameter ϵ as well as the underlying probability distribution according to which $\mathcal{K}$ randomly chooses its noise are allowed to be public knowledge, yet such a mechanism is immune to background knowledge attacks. The differential privacy model is thus viewed by some as perhaps the only acceptable definition of privacy. As an example, Narayanan and Shmatikov [51] argue, in one fell swoop, that none of the "release-and-forget" approaches, of which pseudonymization is one, is capable of providing the privacy level that, at least in principle, interactive differential privacy makes possible. They do, however, immediately acknowledge the impracticality of exclusive usage of interactive differential privacy, due to its prohibitive costs of designing programming interface for queries, budgeting for server resources, etc. Another school of thought, such as Clifton and Tassa [52] and Mohammed et al. [53], makes a strong case that techniques for sanitizing data in some way, prior to its release, are here to stay. The debate on whether to release the entire sanitized data or just sanitized answers to statistical queries is currently ongoing.

This paper does not attempt to take any sides in this debate. As already mentioned in Section I, we are not advocating pseudonymization over any of the other privacy-enhancing techniques. Our work stems from the observation that it is already a widely adopted practice, even legally declared adequate by some privacy laws, yet an accurate way to measure the amount of anonymity it actually provides, is still an unsolved, though already identified, research problem. This paper plugs that research gap.

B. Data Sanitizing Techniques

Several privacy-protection techniques already exist, such as k-anonymity, l-diversity, and t-closeness, each of which attempts to obfuscate, in its own way, the true association between actual data items and released ones. They typically sanitize data based on similarity or diversity of data items, and are thus data-dependent, release-and-forget approaches that do not take into account any background knowledge available to an adversary. Although their underlying privacy parameters, like k, l, and t, are sometimes called "metrics" in the literature, these metrics are simply a measure of the extent to which individuals are guaranteed to be hidden among others in the sanitized data. This guarantee, often achieved by employing anonymity operations like generalization and suppression, is invariably provided at the expense of some utility of the sanitized data.

Pseudonymization, albeit much simpler, is another such release-and-forget technique that does not take the adversary's background knowledge into account. However, by not incorporating any operations like generalization or suppression, it does not suffer from loss of data utility.

C. Scope of Our Metrics

The metrics and heuristic we present in this paper are for measuring the amount of anonymity that remains *after* an attack has been carried out on data sanitized by the pseudonymization technique. In particular, we consider background knowledge attacks that result in some doubly-stochastic probability matrix. Although we develop our metrics and heuristic in the setting of data privacy, pseudonymization-like techniques and probabilistic attacks are also well-recognized in several other domains, such as location privacy [12], [29], trajectory privacy [30], [31], and anonymous communications [15]. Our results are applicable to

these domains as well. While the attacks in each such domain employ domain-specific background knowledge, they lead to, in all cases, a doubly-stochastic probability matrix. As our work starts from that point, it is independent of the actual domain-specific underlying attack method.

As an example of an immediate application, Corollary 1 at the end of Section V-B reveals that a privacy breach in any pseudonymized dataset is the norm. This shows that the widely publicized privacy breaches that led to the resignation of AOL's Chief Technology Officer and a lawsuit on Netflix, as mentioned in Section I, would have been expected had our results been publicly known at that time.

VII. Conclusions

There is growing demand in modern organizations to release their data to external parties, for tasks such as mining useful information from it or complying with governmental data release regulations, while preserving the privacy of individuals or other entities contained in the data. Several techniques for sanitizing such data prior to its release exist, each with its own strengths and weaknesses. In this paper, we study the technique of pseudonymization, which is based upon substituting potentially identifying attribute values in the data, like social security numbers, by fictitious pseudonyms. Specifically, we develop a method for measuring the extent to which the supposedly secret associations, i.e. the perfect matching employed by the data owner, between the actual attribute values and their pseudonyms are still hidden in the aftermath of a probabilistic attack. Our method assists the data owner to decide whether or not the sanitization level of the data is safe enough for a release.

Two methods for a similar task already exist in the literature. However, both of these methods are limited to measuring anonymity only after infeasibility attacks, which is a finite subclass of the uncountably infinite class of probabilistic attacks. Lakshmanan et al. [21] did identify a need for a method for probabilistic attacks, but left that for future work.

The main contribution of our paper is a method to measure anonymity in the wake of any probabilistic attack. Thus, the scope of our work is much wider than of those mentioned above, and subsumes the scope of these earlier works. We develop two different metrics for our purpose. Our first metric, Δ, is a rough metric that first appeared in a preliminary work [23] in the setting of anonymous communications. The second metric, Ψ, is better, as it is more accurate. Both metrics are hard to compute, likely not even in polynomial-time of the size of the dataset. We therefore also develop an efficient heuristic, $\mathcal{H}$, that can be computed in just linear-time, and show that it produces fairly close approximations to Ψ.

Pseudonymization-like techniques and probabilistic attacks are not limited to datasets, but occur in several other domains, such as location privacy [12], [29], trajectory privacy [30], [31], and anonymous communications [15]. Our results are applicable to these domains as well.

Our method measures the effectiveness of pseudonymization in thwarting an attacker's attempt at *pinpointing* the unique matching M_O employed by the data owner. It has already been observed, as in Gierlichs et al. [54], Bagai et al. [55], and Lakshmanan et al. [21], that sometimes the attacker has a more modest goal, especially when the same confidential information is shared by multiple objects. As an example, suppose the attacker has determined that patient names Jessica and Steven in X have been mapped by the data owner, in some order, to the pseudonyms a and b in Y, and the data already shows that both a and b suffer from the acquired immunodeficiency syndrome (AIDS). Then, although the attacker is still unsure of which individual, Jessica or Steven, has been pseudonymized as a or b, he has determined with full certainty that they both have AIDS. Lakshmanan et al. [21] address this by developing a method for determining anonymity of *itemsets*, i.e. subsets of X. Again, their method is limited to infeasibility attacks, and we leave development of a method for probabilistic attacks that takes itemsets into account as future work.

Finally, several variants of the one-to-one pseudonymization technique studied here have already been identified, and some even employed. Some examples are mentioned in Pfitzmann and Hansen [8], such as group and transferable pseudonyms. A *group* pseudonym in Y can be assigned by the data owner to several items in X, inducing its own anonymity set. A *transferable* pseudonym in Y is not always associated by the data owner to the same item in X, especially across multiple releases of the same dataset, thereby affecting anonymity. Combinations of such variants are possible too. A study of the anonymity levels provided by such variants is a natural extension of our work.

References

[1] L. Sweeney, "k-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[2] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "ℓ-diversity: Privacy beyond k-anonymity," in *Proc. 22nd IEEE Int. Conf. Data Eng. (ICDE)*, Apr. 2006, pp. 24–35.

[3] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Proc. 23rd IEEE Int. Conf. Data Eng. (ICDE)*, Apr. 2007, pp. 106–115.

[4] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. ACM SIGMOD Conf. Manage. Data*, 2000, pp. 439–450.

[5] B.-C. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala, "Privacy-preserving data publishing," *Found. Trends Databases*, vol. 2, nos. 1–2, pp. 1–167, 2009.

[6] B. Fung, K. Wang, A. Fu, and P. Yu, *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. Boca Raton, FL, USA: CRC Press, 2011.

[7] C. Aggarwal and P. Yu, Eds., *Privacy-Preserving Data Mining: Models and Algorithms*. New York, NY, USA: Springer, 2008.

[8] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management," Techn. Univ. Dresden, Dresden, Germany Tech. Rep. 0.34, 2010. [Online]. Available: http://dud.inf.tu-dresden.de/Anon_Terminology.shtml

[9] A. Konheim, *Computer Security and Cryptography*. Hoboken, NJ, USA: Wiley, 2007.

[10] T. Neubauer and M. Kolb, "A legal evaluation of pseudonymization approaches," *Int. J. Adv. Secur.*, vol. 2, nos. 2–3, pp. 190–202, 2009.

[11] D. Abouakil, *Pseudonymization of Medical Images: Development and Implementation of a Method for the Pseudonymized Processing of DICOM Images*. Saarbrücken, Germany: VDM Verlag, 2009.

[12] A. Beresford and F. Stajano, "Mix zones: User privacy in location-aware services," in *Proc. IEEE Workshop Pervasive Comput. Commun. Secur. (PerSec)*, Mar. 2004, pp. 127–131.

[13] F. Giannotti, L. Lakshmanan, A. Monreale, D. Pedreschi, and H. Wang, "Privacy-preserving mining of association rules from outsourced transaction databases," *IEEE Syst. J.*, vol. 7, no. 3, pp. 385–395, Mar. 2013.

[14] F. Kerschbaum, "Distance-preserving pseudonymization for timestamps and spatial data," in *Proc. 6th ACM Workshop Privacy Electron. Soc.*, 2007, pp. 68–71.
[15] G. Danezis and C. Troncoso, "Vida: How to use Bayesian inference to de-anonymize persistent communications," in *Proc. 9th Int. Privacy Enhancing Technol. Symp. (PETS)*, vol. 5672. 2009, pp. 56–72.
[16] C. Rottondi, G. Mauri, and G. Verticale, "A data pseudonymization protocol for smart grids," in *Proc. IEEE Online Conf. Green Commun.*, Sep. 2012, pp. 68–73.
[17] AOL. (2006). *AOL Search Data Leak*. [Online]. Available: http://en.wikipedia.org/wiki/AOL_search_data_leak
[18] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proc. 29th IEEE Symp. Secur. Privacy (SP)*, May 2008, pp. 111–125.
[19] Netflix. *Netflix Prize*. 2006. [Online]. Available: http://en.wikipedia.org/wiki/Netflix_Prize
[20] M. Edman, F. Sivrikaya, and B. Yener, "A combinatorial approach to measuring anonymity," in *Proc. IEEE Int. Conf. Intell. Secur. Inf.*, May 2007, pp. 356–363.
[21] L. Lakshmanan, R. Ng, and G. Ramesh, "On disclosure risk analysis of anonymized itemsets in the presence of prior knowledge," *ACM Trans. Knowl. Discovery Data*, vol. 2, no. 3, pp. 13-1–13-44, 2008.
[22] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948.
[23] R. Bagai, H. Lu, R. Li, and B. Tang, "An accurate system-wide anonymity metric for probabilistic attacks," in *Proc. 11th Int. Privacy Enhancing Technol. Symp. (PETS)*, vol. 6794. Jul. 2011, pp. 117–133.
[24] L. Valiant, "The complexity of computing the permanent," *Theor. Comput. Sci.*, vol. 8, no. 2, pp. 189–201, 1979.
[25] L. Lakshmanan, R. Ng, and G. Ramesh, "To do or not to do: The dilemma of disclosing anonymized data," in *Proc. ACM SIGMOD Conf. Manage. Data*, 2005, pp. 61–72.
[26] A. Asratian, T. Denley, and R. Häggkvist, *Bipartite Graphs and Their Applications*. Cambridge, U.K.: Cambridge Univ. Press, 1998.
[27] C. Watanabe, T. Amagasa, and L. Liu, "Privacy risks and countermeasures in publishing and mining social network data," in *Proc. 7th Int. ICST Conf. Collaborative Comput. Netw., Appl. Worksharing (CollaborateCom)*, 2011, pp. 55–66.
[28] S. Ji, W. Li, M. Srivatsa, and R. Beyah, "Structural data deanonymization: Quantification, practice, and implications," in *Proc. 21st ACM Conf. Comput. Commun. Secur. (CCS)*, 2014, pp. 1040–1053.
[29] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2011, pp. 247–262.
[30] S. Gao, J. Ma, W. Shi, G. Zhan, and C. Sun, "TrPF: A trajectory privacy-preserving framework for participatory sensing," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 6, pp. 874–887, Jun. 2013.
[31] K. Mano, K. Minami, and H. Maruyama, "Privacy-preserving publishing of pseudonym-based trajectory location data set," in *Proc. 2nd IEEE Int. Workshop Secur. Mobile Appl.*, Sep. 2013, pp. 615–624.
[32] G. Egorychev, "The solution of van der Waerden's problem for permanents," *Adv. Math.*, vol. 42, no. 3, pp. 299–305, 1981.
[33] R. Bapat and T. Raghavan, *Nonnegative Matrices and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 1997.
[34] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific J. Math.*, vol. 21, no. 2, pp. 343–348, 1967.
[35] B. Kalantari and L. Khachiyan, "On the complexity of nonnegative matrix scaling," *Linear Algebra Appl.*, vol. 240, pp. 87–103, 1996.
[36] N. Linial, A. Samorodnitsky, and A. Wigderson, "A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents," *Combinatorica*, vol. 20, no. 4, pp. 545–568, 2000.
[37] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Proc. 2nd Privacy Enhancing Techol. Workshop*, vol. 2482. 2002, pp. 41–53.
[38] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proc. 2nd Privacy Enhancing Technol. Workshop*, vol. 2482. 2002, pp. 54–68.
[39] J. Kapur, *Maximum Entropy Models in Science and Engineering*, 2nd ed. New Delhi, India: New Age Int. Publishers, 2009.
[40] H. Minc, *Permanents: 006 Encyclopedia of Mathematics and Its Applications*, vol. 6. Reading, MA, USA: Addison-Wesley, 1978.
[41] M. Jerrum and U. Vazirani, "A mildly exponential approximation algorithm for the permanent," *Algorithmica*, vol. 16, pp. 392–401, 1996.
[42] S. Chien, L. Rasmussen, and A. Sinclair, "Clifford algebras and approximating the permanent," *J. Comput. Syst. Sci.*, vol. 67, no. 2, pp. 263–290, 2003.
[43] M. Jerrum, A. Sinclair, and E. Vigoda, "A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries," *J. ACM*, vol. 51, no. 4, pp. 671–697, 2004.
[44] J. Armstrong, ed. *Principles of Forecasting: A Handbook for Researchers and Practitioners*. Norwell, MA, USA: Kluwer, 2001.
[45] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
[46] M. Shcherbakov, A. Brebels, N. Shcherbakova, A. Tyukov, T. Janovsky, and V. Kamaev, "A survey of forecast error measures," *World Appl. Sci. J.*, vol. 24, pp. 171–176, Sep. 2013.
[47] N. Chen, Z. Qian, X. Meng, and I. Nabney, "Short-term wind power forecasting using Gaussian processes," in *Proc. 23rd Int. Joint Conf. Artif. Intell. (IJCAI)*, 2013, pp. 2790–2796.
[48] C. Dwork, "Differential privacy," in *Automata, Languages and Programming* (Lecture Notes in Computer Science), vol. 4052, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Heidelberg, Germany: Springer, 2006, pp. 1–12.
[49] F. Dankar and K. E. Emam, "Practicing differential privacy in health care: A review," *Trans. Data Privacy*, vol. 6, no. 1, pp. 35–67, 2013.
[50] C. Dwork, M. Naor, O. Reingold, G. Rothblum, and S. Vadhan, "On the complexity of differentially private data release: Efficient algorithms and hardness results," in *Proc. 41st ACM Symp. Theory Comput. (ACM-STOC)*, 2009, pp. 381–390.
[51] A. Narayanan and V. Shmatikov, "Myths and fallacies of 'personally identifiable information,"' *Commun. ACM*, vol. 53, no. 6, pp. 24–26, 2010.
[52] C. Clifton and T. Tassa, "On syntactic anonymity and differential privacy," *Trans. Data Privacy*, vol. 6, no. 2, pp. 161–183, 2013.
[53] N. Mohammed, B. Fung, P. Hung, and C.-K. Lee, "Anonymizing healthcare data: A case study on the blood transfusion service," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2009, pp. 1285–1294.
[54] B. Gierlichs, C. Troncoso, C. Diaz, B. Preneel, and I. Verbauwhede, "Revisiting a combinatorial approach toward measuring anonymity," in *Proc. 7th ACM Workshop Privacy Electron. Soc.*, 2008, pp. 111–116.
[55] R. Bagai, B. Tang, and E. Kim, "Effectiveness of probabilistic attacks on anonymity of users communicating via multiple messages," *IEEE Syst. J.*, vol. 7, no. 2, pp. 199–210, Feb. 2013.

Rajiv Bagai received the B.S. degree in computer science from the Birla Institute of Technology and Science, India, and the M.S. and Ph.D. degrees in computer science from the University of Victoria, Canada. He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, Wichita State University, USA. His current research is in Web and data anonymity, but in the past he has been involved in logic programming and paraconsistent databases.

Nafia Malik received the B.S. degree in computer science and engineering from the Military Institute of Science and Technology, Bangladesh, and the M.S. degree in computer networking from Wichita State University, USA. She is currently a Software Engineer with the Cyber Security Group, Panasonic Automotive Systems Company of America, USA. Her research interests include software security, computer networking, web anonymity, and privacy.

Murtuza Jadliwala received the B.E. degree in computer engineering from Mumbai University, India, and the M.S. and Ph.D. degrees in computer science from the State University of New York at Buffalo, USA. He is currently an Assistant Professor with the Department of Electrical Engineering and Computer Science, Wichita State University, USA. His current research focuses on overcoming security threats and privacy challenges in networked and cyber-physical systems.