k-Anonymization Revisited

Aristides Gionis [#1], Arnon Mazza [*2], Tamir Tassa [*3]

[#] *Yahoo! Research, Barcelona, Spain*
[1] gionis@yahoo-inc.com

[*] *Division of Computer Science, The Open University Ra'anana, Israel*
[2] arnon.mazza@verint.com
[3] tamirta@openu.ac.il

***Abstract*— In this paper we introduce new notions of k-type anonymizations. Those notions achieve similar privacy goals as those aimed by Sweenie and Samarati when proposing the concept of k-anonymization: an adversary who knows the public data of an individual cannot link that individual to less than k records in the anonymized table. Every anonymized table that satisfies k-anonymity complies also with the anonymity constraints dictated by the new notions, but the converse is not necessarily true. Thus, those new notions allow generalized tables that may offer higher utility than k-anonymized tables, while still preserving the required privacy constraints. We discuss and compare the new anonymization concepts, which we call $(1,k)$-, (k,k)- and global $(1,k)$-anonymizations, according to several utility measures. We propose a collection of agglomerative algorithms for the problem of finding such anonymizations with high utility, and demonstrate the usefulness of our definitions and our algorithms through extensive experimental evaluation on real and synthetic datasets.**

I. Introduction

As data mining algorithms are becoming ubiquitous and as data are continuously collected and shared within organizations, *privacy-preserving data mining* [5], [20] has been proposed as a paradigm of exercising data mining while protecting the privacy of individuals.

One of the most well-studied methods of privacy-preserving data mining is called k-anonymization [3], [6], [14], [16], [19]. Consider an organization (e.g., a hospital) that holds information on a set of individuals (e.g., hospitalized patients). That information is kept in a database table where each record holds the information of a single individual. The personal information consists of several attributes, some of which are private (e.g., `illness`) and some are public (e.g., `name`, `date_of_birth`, `zipcode`) and can be found in public databases such as the voter register. The organization, for the purposes of data mining or other types of statistical research, needs to publish the data, but at the same time it is committed to protect the private information of the individuals. The method of k-anonymization suggests to modify the values of the public attributes of the data so that if the database table is projected on the subset of the public attributes (a.k.a. *quasi-identifiers*), each record of the table becomes identical with at least $k-1$ other records.

The values of the database table are modified via the operations of *generalization* or *suppression*, and typically, a *cost function* is used to measure the amount of information lost by modifying the data. Clearly, by reducing the amount of information lost in the process of k-anonymizing a table, we increase the utility of the released table for the purposes of data mining. Hence, the objective is to modify the table entries so that the table becomes k-anonymized and the information loss is minimized.

Our focus in this paper is to explore different notions of k-type anonymizations that lead to anonymized data with higher utility. To demonstrate the basic idea, consider first the case where an adversary knows the public information of a single individual. Then instead of obtaining a fully k-anonymized table, we may generalize the table entries so that the public data of every individual is consistent with the public data in at least k records of the released table. We call such tables $(1,k)$-*anonymized*. The notion of $(1,k)$-anonymity is a *relaxation* of k-anonymity in the sense that every k-anonymized table is also $(1,k)$-anonymized, but the converse is not necessarily true. Thus, the optimal utility of any data that is $(1,k)$-anonymized is at least as large as the optimal utility of the same data that is k-anonymized.

Another notion that we introduce is that of $(k,1)$-anonymity. Any record in a $(k,1)$-anonymized table is consistent with at least k original records. Clearly, every k-anonymization is also a $(k,1)$-anonymization, and thus $(k,1)$-anonymity may give generalized tables with higher utility.

It turns out that $(1,k)$-anonymity and $(k,1)$-anonymity are too weak, as we demonstrate later. Hence, we proceed to introduce the stronger notion of (k,k)-*anonymity*. A table is (k,k)-anonymized if it is both $(1,k)$- and $(k,1)$-anonymized. Such tables seem to provide similar security to that of k-anonymized tables, in the typical scenario where the adversary has only full knowledge on some of the individuals. Since (k,k)-anonymity is also a relaxation of k-anonymity, adopting that notion as the security goal may be rewarded with higher utility.

However, if the adversary knows the exact subset of the population that is represented in the database, and she also knows the public data of *all* of those individuals, (k,k)-anonymity may offer lower security than k-anonymity. For such scenarios we present the final notion of *global* $(1,k)$-*anonymity*. Global $(1,k)$-anonymity offers the same security as k-anonymity. Namely, even with complete knowledge of all public data in the database it is not possible to link any

individual to less than k records of the anonymized table. As global $(1,k)$-anonymity is also a relaxation of k-anonymity, it too may result with tables that have better utility than k-anonymized tables.

While (k,k)-anonymity may be achieved with a reasonable computational cost, global $(1,k)$-anonymity entails a larger computational toll. However, the adversarial model that poses a threat on (k,k)-anonymity and justifies the notion of global $(1,k)$-anonymity seems to be unrealistic in typical scenarios. Hence, we believe that in practice (k,k)-anonymity may serve as a good alternative to both global $(1,k)$-anonymity and k-anonymity.

In this paper we are making the following contributions.

- We introduce new notions of k-type anonymizations that lead to anonymized tables with higher utility. We characterize the relations among the new anonymity notions and the original notion of k-anonymity, and discuss the underlying security assumptions.
- We propose a collection of agglomerative algorithms for the problem of finding high-utility anonymizations that are consistent with our new anonymity concepts.
- We demonstrate the usefulness of our definitions and our proposed algorithms through extensive experimental evaluation on real and synthetic datasets.

The rest of the paper is organized as follows. In Section II we discuss related work. In Section III we formally define the basic concepts, and in Section IV we introduce the new anonymity notions. In Section V we propose algorithms for anonymizing data according to the standard k-anonymity notion, as well as according to the new notions. In Section VI we describe our experiments, and finally we conclude in Section VII.

II. Related Work

The objective of protecting the privacy of individuals represented in databases has been formulated by Dalenius [8] already in 1977. Since then, many approaches have been suggested for finding the right path between data hiding and data disclosure. Such approaches include query auditing [13], output perturbation [7], secure multi-party computation [4], and data sanitization [5], [9].

One of the recent approaches, proposed by Samarati and Sweeney [18], [19], is k-anonymization. Meyerson and Williams [16] introduced the problem of transforming a database table using suppressions so that the k-anonymity property is satisfied and the amount of information loss due to the suppression operations is minimized. They showed that this problem is NP-hard and they devised two approximation algorithms: one with running time $O(n^{2k})$ and approximation ratio $O(k \ln k)$, and one with fully polynomial running time and approximation ratio $O(k \ln n)$. Aggarwal et al. [2], [3] extended the setting of suppressions-only by allowing more general rules for generalizing data entries and they devised a polynomial $O(k)$-approximation algorithm.

The information loss function proposed by Aggarwal et al. [2], [3] is defined as a *tree measure* and it is a generalization of the function considered by Meyerson and Williams [16]. In [10], three *entropy-based* functions are suggested for measuring the information loss. Those measures are more general than the tree measure, as they apply to any type of generalization, and they capture more accurately the information loss due to anonymization. An $O(\ln k)$-approximation algorithm is presented in [10] for the problem of optimal k-anonymity with respect to two of the entropy-based measures, as well as for the tree measure. We review the basic entropy-based measure in Section IV.

Other information loss measures were used in previous studies. The LM measure [11], [17] is a more precise version of the tree measure of Aggarwal et al. The CM measure [11] and the DM measure [6] were also used as cost metric measures. Our notions of k-type anonymity are independent of the underlying cost measure. In our experiments, we use the basic entropy measure of [10], as a representative of the three entropy-based measures that were presented there, and the LM measure, which seems to be the most accurate measure from among the above mentioned measures.

Recently, Aggrawal et al. [1] proposed to anonymize data by first clustering the data records and then publish cluster centers and radii. Our new anonymity notions are independent of the underlying clustering method and, consequently, they may be applied also with these clustering techniques.

LeFevre et al. [14] suggested a k-anonymization algorithm in the model of full-domain generalization, while Bayardo and Agrawal [6] proposed an optimal algorithm in the model of global recoding. Those algorithms are not directly comparable to our present work since we consider the model of local recoding, in order to optimize the utility of the anonymized data. Consequently, in our experiments we compare our algorithms to the algorithm of Aggarwal et al. [2], [3], since to the best of our knowledge it is the leading practical algorithm for k-anonymity in the local-recoding model. Our agglomerative algorithm is similar in flavor to the bottom-up algorithm presented by Xu et al. [22]. However, we also extend this bottom-up algorithm by considering different utility measures and exploring alternative merging strategies.

In a slightly different line of research, Machanavajjhala et al. [15] proposed the concept of ℓ-diversity, as a necessary enhancement to k-anonymity. We believe that ℓ-diversity fits also in our framework, but we have left the investigation of this topic for future research.

Similar to the spirit of our paper, but not directly comparable, are the recent works of Kifer and Gehrke [12], and Xiao and Tao [21]. Both works aim at improving the utility of the anonymized data. Kifer and Gehrke [12] suggest publishing *many marginals* of the data instead of a single k-anonymous l-diverse table, in order to obtain better utility while respecting similar privacy properties. Xiao and Tao [21] propose publishing the table with all non-sensitive attributes unaltered, while the sensitive attribute in each record is replaced by a label of an ℓ-diverse group of sensitive attribute values. In addition, they publish the distribution of the sensitive attribute values within each such group.

III. PRELIMINARIES

Consider a database that holds information on individuals in some population $U = \{u_1, \ldots, u_n\}$. Each individual is described by a collection of r public attributes (also known as *quasi-identifiers*), $A_1, \ldots, A_r$, and s private attributes, $Z_1, \ldots, Z_s$. Each of the attributes consists of several possible values: $A_j = \{a_{j,\ell} : 1 \leq \ell \leq m_j\}$, $1 \leq j \leq r$, and $Z_j = \{z_{j,\ell} : 1 \leq \ell \leq n_j\}$, $1 \leq j \leq s$. For example, if A_j is the attribute `gender` then $A_j = \{\texttt{M},\texttt{F}\}$, while if A_j is the attribute `age`, then it is a bounded nonnegative natural number.

The public database D holds all publicly available information on the individuals in U:

$$D = \{R_1, \ldots, R_n\}, \text{ with } R_i \in A_1 \times \cdots \times A_r,\ 1 \leq i \leq n. \tag{1}$$

The corresponding private database D' holds the private information,

$$D' = \{S_1, \ldots, S_n\}, \text{ with } S_i \in Z_1 \times \cdots \times Z_s, 1 \leq i \leq n. \tag{2}$$

The complete database is the concatenation of those two databases, $D \| D' = \{R_1 \| S_1, \ldots, R_n \| S_n\}$. We refer hereinafter to the tuples R_i and S_i, $1 \leq i \leq n$, as public and private records, respectively. We denote the j-th component of the record R_i by $R_i(j)$. Also, for any set A we let $\mathcal{P}(A)$ denote its power set.

Definition 3.1: Let A_j, $1 \leq j \leq r$, be finite sets and let $\overline{A}_j \subseteq \mathcal{P}(A_j)$ be a collection of subsets of A_j. A mapping $g : A_1 \times \cdots \times A_r \to \overline{A}_1 \times \cdots \times \overline{A}_r$ is called a generalization if for every $(b_1, \ldots, b_r) \in A_1 \times \cdots \times A_r$ and $(B_1, \ldots, B_r) = g(b_1, \ldots, b_r)$, it holds that $b_j \in B_j$, $1 \leq j \leq r$.

As an example consider a database D with two attributes, `age` (A_1) and `zipcode` (A_2). A valid generalization of a record $R_i = (34, 68423) \in D$ can be $g((34, 68423)) = (\{30, \ldots, 39\}, \{68400, \ldots, 68499\})$.

Definition 3.1 refers to generalizations of single records. We now define generalizations of an entire database.

Definition 3.2: Let $D = \{R_1, \ldots, R_n\}$ be a database with public attributes $A_1, \ldots, A_r$, $\overline{A}_1, \ldots, \overline{A}_r$ be corresponding collections of subsets, and $g_i : A_1 \times \cdots \times A_r \to \overline{A}_1 \times \cdots \times \overline{A}_r$ be corresponding generalization operators. Let $\overline{R}_i := g_i(R_i)$ be the generalization of the record R_i, $1 \leq i \leq n$. Then $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ is a generalization of D.

It is important to note that Definition 3.2 refers to *local recoding*, in the sense that a different mapping g_i may be applied to different records. This is in contrast with *global recoding* where the same mapping g must be applied to all records. Local recoding is more flexible, hence it offers higher utility. For example, assume that one of the attributes in the database is `age`, and that there exist several records with the value 34 under that attribute. Then it is allowed to leave that value unchanged in some of those records, replace it with the range $\{30, \ldots, 39\}$ in some other records, and replace it with a different range, say $\{20, \ldots, 49\}$, or even totally suppress it in other records.

Finally, we define *consistency* between records of the original and the generalized database.

Definition 3.3: Let $R_i \in D$ be an original record and $\overline{R}_j \in g(D)$ be a generalized record. We say that $\overline{R}_j$ generalizes R_i, or, equivalently, that they are consistent, if $R_i(h) \in \overline{R}_j(h)$ for all $1 \leq h \leq r$.

IV. k-ANONYMIZATION REVISITED

We begin this section by reviewing the notion of k-anonymity as it is used in the recent literature [3], [6], [10], [14], [16]. We then introduce the new notions of k-type anonymity and discuss them and their interrelations. All of those notions are relaxations of k-anonymity, hence they allow greater utility. The commonly used notion of k-anonymity [3], [6], [14], [16] is defined as follows:

Definition 4.1 (k-anonymity): A k-anonymization of a database $D = \{R_1, \ldots, R_n\}$ is a generalization $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ where for all $1 \leq i \leq n$, there exist indices $1 \leq i_1 < i_2 < \cdots < i_{k-1} \leq n$, all of which are different from i, such that $\overline{R}_i = \overline{R}_{i_1} = \cdots = \overline{R}_{i_{k-1}}$.

The objective in this context is to generalize a given database until it becomes k-anonymized, while incurring a minimal loss of information. Let Π be a measure of the amount of information that is lost by replacing a database D with a corresponding generalization $g(D)$. Then the problem of k-anonymization is as follows.

Definition 4.2 (k-anonymization problem): Let $D = \{R_1, \ldots, R_n\}$ be a database with public attributes A_j, $1 \leq j \leq r$. Given collections of attribute values, $\overline{A}_j \subseteq \mathcal{P}(A_j)$, and a measure of information loss Π, find a corresponding k-anonymization, $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$, where $\overline{R}_i \in \overline{A}_1 \times \cdots \times \overline{A}_r$, that minimizes $\Pi(D, g(D))$.

The measure of loss of information Π took several forms in previous studies. Meyerson and Williams [16] considered the case of generalization by suppression, and their measure simply counted the number of suppressed entries. Aggarwal et al. [2], [3] used a more general model, in which every single value may be replaced by a node in a hierarchy tree, and the corresponding cost is proportional to the level in the hierarchy that was selected. In this paper, we consider a more general and more accurate entropy-based measure of loss of information, which was proposed and studied in [10]. We proceed to review this measure.

The public database $D = \{R_1, \ldots, R_n\}$ induces a probability distribution for each of the public attributes. Let X_j, $1 \leq j \leq r$, denote the value of the attribute A_j in a randomly selected record from D. Then

$$\Pr(X_j = a) = \frac{\#\{1 \leq i \leq n : R_i(j) = a\}}{n}.$$

Let B_j be a subset of A_j. The conditional entropy $H(X_j|B_j)$ is defined as

$$H(X_j|B_j) = -\sum_{b\in B_j} \Pr(b|B_j)\log_2 \Pr(b|B_j)\,,$$

where

$$\Pr(b|B_j) = \frac{\#\{1\le i\le n : R_i(j)=b\}}{\#\{1\le i\le n : R_i(j)\in B_j\}},\ b\in B_j.$$

The entropy-based information loss function for generalization is now defined as follows.

Definition 4.3: Let $D=\{R_1,\ldots,R_n\}$ be a database having public attributes $A_1,\ldots,A_r$, and let X_j be the random variable that equals the value of the j-th attribute A_j, $1\le j\le r$, in a randomly selected record from D. Then if $g(D)=\{\overline{R}_1,\ldots,\overline{R}_n\}$ is a generalization of D,

$$\Pi_E(D,g(D)) = \frac{1}{nr}\cdot\sum_{i=1}^{n}\sum_{j=1}^{r} H(X_j|\overline{R}_i(j)) \qquad (3)$$

is the entropy measure of the loss of information caused by generalizing D into $g(D)$.

Another measure that we use in our experiments is the LM measure [11], [17]. The cost per each table entry is a number between 0 (no generalization at all) and 1 (total suppression) that penalizes the generalization that was made in that entry, and the overall cost is the average over the costs of all table entries:

$$\Pi_{\mathrm{LM}}(D,g(D)) = \frac{1}{nr}\cdot\sum_{i=1}^{n}\sum_{j=1}^{r}\frac{|\overline{R}_i(j)|-1}{|A_j|-1} \qquad (4)$$

We now proceed to introduce our novel notions of k-type anonymity. Those notions rely on the concept of *consistency*, that was defined in Definition 3.3.

Definition 4.4: Let $D=\{R_1,\ldots,R_n\}$ be a table and $g(D)=\{\overline{R}_1,\ldots,\overline{R}_n\}$ be a corresponding generalization. Then

- $g(D)$ is a $(1,k)$-anonymization of D if each record in D is consistent with at least k records in $g(D)$.
- $g(D)$ is a $(k,1)$-anonymization of D if each record in $g(D)$ is consistent with at least k records in D.
- $g(D)$ is a (k,k)-anonymization of D if it is both a $(1,k)$- and a $(k,1)$-anonymization of D.

Correspondingly, we define $\mathcal{A}_D^k$, $\mathcal{A}_D^{(1,k)}$, $\mathcal{A}_D^{(k,1)}$, and $\mathcal{A}_D^{(k,k)}$ to be the collections of all of the k-, $(1,k)$-, $(k,1)$- and (k,k)-anonymizations of the database D, respectively.

We proceed to state and prove the interrelations between these four notions of k-type anonymity.

Proposition 4.5: For a given table D, let the collections $\mathcal{A}_D^k$, $\mathcal{A}_D^{(1,k)}$, $\mathcal{A}_D^{(k,1)}$, and $\mathcal{A}_D^{(k,k)}$ be as in Definition 4.4. Then

$$\mathcal{A}_D^k \subsetneqq \mathcal{A}_D^{(k,k)} \subsetneqq \mathcal{A}_D^{(1,k)}, \mathcal{A}_D^{(k,1)}, \qquad (5)$$

and

$$\mathcal{A}_D^{(1,k)} \setminus \mathcal{A}_D^{(k,1)} \neq \emptyset, \quad \mathcal{A}_D^{(k,1)} \setminus \mathcal{A}_D^{(1,k)} \neq \emptyset. \qquad (6)$$

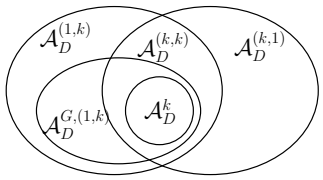

Fig. 1. Interrelations between the five classes of k-type anonymizations.

(See Figure 1.)

Proof: As all the inclusions in (5) are straightforward, it remains only to exemplify the inequalities in (5) and (6). We demonstrate those inequalities for the case $k=2$, but those examples may be easily extended to any k.

D	2-anon	(1,2)-anon	(2,1)-anon	(2,2)-anon
1, 3	{1,2}, {3,4}	1, 3	1, {3,4}	1, {3,4}
1, 4	{1,2}, {3,4}	{1,2}, {3,4}	{1,2}, 4	{1,2}, {3,4}
2, 4	{1,2}, {3,4}	{1,2}, 4	{1,2}, 4	{1,2}, 4

The above table D (having two attributes and three records) is shown along with four anonymizations, from $\mathcal{A}_D^2$, $\mathcal{A}_D^{(1,2)}$, $\mathcal{A}_D^{(2,1)}$, and $\mathcal{A}_D^{(2,2)}$, resp. The second generalization is in $\mathcal{A}_D^{(1,2)}$ but not in $\mathcal{A}_D^{(2,1)}$ (and, hence, not in $\mathcal{A}_D^{(2,2)}$). The third generalization is in $\mathcal{A}_D^{(2,1)}$ but not in $\mathcal{A}_D^{(1,2)}$ (and, hence, not in $\mathcal{A}_D^{(2,2)}$). The last generalization is in $\mathcal{A}_D^{(2,2)}$ but not in $\mathcal{A}_D^2$. ■

Our anonymity definitions can also be understood via graph terminology, as follows: Let $D=\{R_1,\ldots,R_n\}$ be a table and $g(D)=\{\overline{R}_1,\ldots,\overline{R}_n\}$ be a corresponding generalization. This pair of tables defines a bipartite graph $V_{D,g(D)}$ on the set of nodes $D\cup g(D)$ where an edge connects $R_i\in D$ with $\overline{R}_j\in g(D)$ if and only if the two records are consistent. With this formulation, $\mathcal{A}_D^{(1,k)}$ (respectively, $\mathcal{A}_D^{(k,1)}$, or $\mathcal{A}_D^{(k,k)}$) is the collection of all generalizations $g(D)$ for which every node in D (respectively $g(D)$, or $D\cup g(D)$) in the graph $V_{D,g(D)}$ has degree at least k. This formulation in terms of the underlying bipartite graph, gives rise to yet another notion of k-type anonymization.

Definition 4.6: Let D and $g(D)$ be a table and its generalization, and let $V_{D,g(D)}$ be the corresponding bipartite graph. A record $\overline{R}\in g(D)$ is called a match of $R\in D$ if $(R,\overline{R})$ is an edge and it may be completed to a perfect matching in $V_{D,g(D)}$. If all records $R\in D$ have at least k matches in $g(D)$, then $g(D)$ is called a global $(1,k)$-anonymization of D.

The relation between the new anonymization class and the previous ones is given in the following Proposition.

Proposition 4.7: Let $\mathcal{A}_D^{G,(1,k)}$ denote the collection of all global $(1,k)$-anonymizations of D. Then the relation between the five classes of anonymizations – $\mathcal{A}_D^k$, $\mathcal{A}_D^{(1,k)}$, $\mathcal{A}_D^{(k,1)}$, $\mathcal{A}_D^{(k,k)}$ and $\mathcal{A}_D^{G,(1,k)}$, is as depicted in Figure 1.

A. Discussion

Here we discuss the security of these new notions of k-type anonymity. To that end, we distinguish between two adversaries. The first one knows the public data of all individuals

in the population and the identity of some individuals in the database. The second one knows, in addition to that, what is the subset of the entire population that is represented in the database.

We begin by considering the security of $(k,1)$- and $(1,k)$-anonymity. Both are insecure. Consider a database D and a generalization $g(D)$ of D that satisfies $(k,1)$-anonymity. Namely, each record in $g(D)$ is consistent with at least k records in D. However, it is possible that a record $R \in D$ is consistent with only one record in $g(D)$ – as for example is the case for the first record in the database of the proof of Proposition 4.5. Thus, even the first adversary might be able to reveal the private information of an individual, based on his public information.

Next, consider a $(1,k)$-anonymization $g(D)$ of D. It is true that every record in D is consistent with at least k records in $g(D)$, hence such anonymizations seem to satisfy our privacy goal. However, the following example shows where this notion fails. Assume that $D = \{R_1, \ldots, R_n\}$ and that $\overline{R}^*$ is a generalized record that is consistent with all records in D (e.g., all entries in $\overline{R}^*$ are suppressed). Consider the following generalization $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$, where $\overline{R}_i = R_i$ for all $1 \leq i \leq n-k$ and $\overline{R}_i = \overline{R}^*$ for all $n-k+1 \leq i \leq n$. It is easy to see that $g(D) \in \mathcal{A}_D^{(1,k)}$. Moreover, since most of the records in $g(D)$ were not generalized at all, the information loss $\Pi(D, g(D))$ is very small, for any measure Π. However, such a generalization is completely unacceptable: The private information of most of the individuals represented in D is completely revealed.

The notion of (k,k)-anonymity combines the two previous notions and it seems that it does not suffer from the above mentioned shortcomings of those two notions. The first adversary may link the public data of an individual to no less than k records in the generalized database. Hence, this notion seems to provide the same level of security as that of k-anonymity. As it entails possibly smaller losses of information than k-anonymity, it seems like the method of choice in practical settings.

However, that notion may fail to provide the sought-after level of privacy under the second adversarial assumption. In the full version of this paper we describe an attack that the second adversary may exercise on a (k,k)-anonymized table in order to link the public information of an individual to less than k generalized records. The attack works as follows: Assume that the adversary wishes to find the private data that corresponds to a record $R_i \in D$. The second adversary, who knows both D and $g(D)$, may construct the graph $V_{D,g(D)}$. In that graph, the node that corresponds to R_i is connected to at least k nodes in $g(D)$. The adversary may test each of the neighbors of R_i to see which of them is a match. The generalized record that corresponds to R_i must be one of those matches. But the number of matches, as opposed to the number of neighbors, may be smaller than k. This motivates our definition of global $(1,k)$-anonymity, which is the most secure notion from among the four novel notions of k-type anonymity that we presented here. It is as secure as the original notion of k-anonymity under the second adversarial assumption. Indeed, even the second adversary may not link any individual to less than k records in the generalized database, because there are always at least k possible different matches for the record of that individual, each of which is equally probable. In the next section we show how to convert a (k,k)-anonymized database into a global $(1,k)$-anonymized one.

In many scenarios, the second adversarial model seems unrealistic. Consider, for example, the case of a hospital that publishes a (k,k)-anonymized database of its patients. Even if the adversary knows that a particular individual has been treated in that hospital, and she knows the public information of all individuals in the entire population, she still might not know the exact subset of the population that has been treated in the hospital. Furthermore, in the typical case where the published database represents the unified population of patients in several hospitals, such an adversarial model becomes even less reasonable. Hence, it seems that in most scenarios (k,k)-anonymity provides the same level of security as that aimed by k-anonymity.

In the full version of this paper we discuss an even stronger adversary – one that also has auxiliary knowledge such as the private data of some of the individuals in the database. Herein, we omit further discussion of that adversarial model. To the best of our knowledge, none of the previous studies of k-anonymity addresses the issue of adversarial assumptions. In our opinion, as k-anonymity is a solution to a problem of privacy, a discussion of adversarial assumptions is in order.

V. Algorithms

In this section we describe various algorithms for k-, (k,k)- and global $(1,k)$-anonymization, and compare their performance. The best practical k-anonymization algorithm with a provable approximation guarantee is the one due to Aggarwal et al. [2], [3]. That algorithm, which we call herein *the forest algorithm*, guarantees an approximation ratio of $3k-3$. In practice, however, better results may be obtained by heuristic algorithms. In Section V-A we describe such heuristic algorithms that, as demonstrated later, outperform the forest algorithm. Then, in Section V-B, we describe an algorithm for (k,k)-anonymization, and in Section V-C we describe an algorithm for transforming a (k,k)-anonymization into a global $(1,k)$-anonymization.

A. k-Anonymization

In this section we describe heuristic algorithms for the k-anonymization problem. In Section V-A.1 we present our main algorithm, *the agglomerative algorithm*, and then describe another variant of it, to which we refer as *the modified agglomerative algorithm*. Both algorithms depend upon a definition of distance between subsets of records. In Section V-A.2 we describe four choices of distance functions.

1) The basic agglomerative algorithm: Given a database $D = \{R_1, \ldots, R_n\}$ and an integer $k > 1$, we compute a clustering of D, $\gamma = \{S_1, \ldots, S_m\}$, (namely, $S_i \subset D$, $S_i \cap S_j = \emptyset$ and $\bigcup_{1 \leq i \leq m} S_i = D$) such that $|S_i| \geq k$ for all $1 \leq$

$i \leq m$. The algorithm assumes a distance function, $\text{dist}(\cdot,\cdot)$, between subsets of D, i.e., dist: $\mathcal{P}(D) \times \mathcal{P}(D) \to \mathbb{R}$.

Algorithm 1 Basic algorithm for k-anonymization

Input: Table D, integer k.
Output: Table $g(D)$ that satisfies k-anonymity.

1: For each record $R_i \in D$ create a singleton cluster $\hat{S}_i = \{R_i\}$ and let $\hat{\gamma} = \{\hat{S}_1, \ldots, \hat{S}_n\}$
2: Initialize the output clustering γ to $\emptyset$.
3: **while** $|\hat{\gamma}| > 1$ **do**
4: Find the "closest" two clusters in $\hat{\gamma}$, namely, the two clusters $\hat{S}_i, \hat{S}_j \in \hat{\gamma}$ that minimize $\text{dist}(\hat{S}_i, \hat{S}_j)$.
5: Set $\hat{S} = \hat{S}_i \cup \hat{S}_j$.
6: Remove $\hat{S}_i$ and $\hat{S}_j$ from $\hat{\gamma}$.
7: If $|\hat{S}| < k$ add $\hat{S}$ to $\hat{\gamma}$.
8: Else add $\hat{S}$ to γ.
9: **end while**
(At this stage, $\hat{\gamma}$ has at most one cluster, $\hat{S} = \{R_{i_1}, \ldots, R_{i_\ell}\}$, the size of which is $\ell < k$)
10: For each record R_{i_j}, $1 \leq j \leq \ell$, add that record to the cluster S in γ that minimizes $\text{dist}(\{R_{i_j}\}, S)$.

Our basic agglomerative algorithm, Algorithm 1, starts with singleton clusters and then keeps unifying the two closest clusters until they mature into clusters of size at least k. As it may produce clusters of size greater than k, while it is preferable to have clusters of size k or close to k in order to reduce the clustering anonymization cost, we propose an improved version of the above described algorithm—the modified agglomerative algorithm. Algorithm 2 describes how to replace line 8 of Algorithm 1 in order to achieve that goal. Essentially, before moving a "ripe" cluster $\hat{S}$ to the final clustering γ, we shrink it to a sub-cluster of size k.

Algorithm 2 Modification of line 8 of Algorithm 1

Input: $\hat{S} = \{\hat{R}_1, \ldots, \hat{R}_\ell\}$ where $\ell > k$.

1: **while** $\hat{S}$ has size greater than k **do**
2: For all $1 \leq i \leq \ell$, compute $d_i = \text{dist}(\hat{S}, \hat{S} \setminus \{\hat{R}_i\})$.
3: Find the record $\hat{R}_i$ that maximizes d_i.
4: Remove $\hat{R}_i$ from $\hat{S}$ and add the corresponding singleton cluster $\{\hat{R}_i\}$ to $\hat{\gamma}$.
5: **end while**
6: Place the shrunk cluster $\hat{S}$ (of size k) in γ.

Finally, the clustering of D that is produced by either of the above agglomerative algorithms is translated into a corresponding generalization $g(D)$ as follows: Every record $R_i \in D$ is replaced by the *closure* of the cluster to which R_i belongs, where a closure of a subset of records is the *minimal* generalized record that is consistent with all of them. Since all of the clusters are of size at least k, every generalized record in $g(D)$ is indistinguishable from at least $k-1$ other generalized records. The running time of the agglomerative algorithm is $O(n^2)$.

2) The distance function: A key ingredient in the agglomerative algorithms is the definition of distance between clusters. It is natural to define the distance so that it best fits the cost function of the k-anonymization. We used in our experiments two measures – the entropy measure, (3), and the LM measure, (4). Both take the form $\Pi(D, g(D)) = \frac{1}{n}\sum_{i=1}^{n} c(\overline{R}_i)$, where $c(\overline{R}_i)$ is the corresponding generalization cost of the generalized record $\overline{R}_i$. (I.e., $c(\overline{R}_i) = \frac{1}{r}\sum_{j=1}^{r} H(X_j|\overline{R}_i(j))$ in the case of the entropy measure, and $c(\overline{R}_i) = \frac{1}{r}\sum_{j=1}^{r} \frac{|\overline{R}_i(j)|-1}{|A_j|-1}$ in the case of the LM measure.) Since all records in a given cluster are replaced by the same generalized record, we have

$$\Pi(D, g(D)) = \sum_{S \in \gamma} |S| \cdot d(S)\,, \text{ where } \; d(S) = c(\overline{S})\,. \quad (7)$$

We use hereinafter the above definition of the function $d(\cdot)$ as the *generalization cost* of any subset of records. Given such a subset S, its generalization cost $d(S)$ is the generalization cost $c(\cdot)$ of its closure $\overline{S}$.

We briefly describe below four choices of distance functions that can be used in the basic agglomerative algorithm. A detailed discussion of those distance functions is postponed for the full version of the paper.

Distance function 1. Our first definition of distance between two clusters A and B is:

$$\text{dist}(A, B) = |A \cup B| \cdot d(A \cup B) - |A| \cdot d(A) - |B| \cdot d(B). \quad (8)$$

A property of this distance function is that it usually favors the unification of smaller clusters, thus resulting in a balanced growth of cluster sizes.

Distance function 2. The second function we use is

$$\text{dist}(A, B) = d(A \cup B) - d(A) - d(B). \quad (9)$$

This function may attain negative values, hence, it is not a genuine distance function. However, it still serves our goal as the measure for the price that we pay in terms of loss of information when choosing to unify the clusters A and B. Using function (9) gives rise to unbalanced cluster sizes during the merging process. Namely, a typical behavior is that one cluster grows and evolves to its full size and only then another small cluster starts to evolve to its full size.

Distance function 3. The experimental comparison between the two previous distance functions indicated that an unbalanced formation of clusters is preferable to a balanced one. Using the distance definition

$$\text{dist}(A, B) = \frac{d(A \cup B) - d(A) - d(B)}{\log(|A \cup B|)}, \quad (10)$$

takes that idea one step further. The division by $\log(|A \cup B|)$ gives priority to adding a record to a larger cluster. Our experiments show that it performs slightly better than the function (9).

Distance function 4. The final variant of a distance function that we use is

$$\text{dist}(A, B) = \frac{d(A \cup B)}{d(A) + d(B) + \varepsilon}\,. \quad (11)$$

Given two subsets, A and B, this function returns the factor by which the generalization cost of the union $A \cup B$ increases the sum of the generalization costs of A and B. The additive constant in the denominator is needed for the cases where both A and B are singletons and hence have a zero generalization cost. In our experiments we used $\varepsilon = 0.1$.

We conclude this section by noting that, recently, Nergiz and Clifton [17] devised also an agglomerative clustering algorithm that is similar to our basic algorithm. The distance function that they used is $\text{dist}(A, B) = d(A \cup B) - d(B)$, which is an asymmetric version of our second distance function, (9).

B. (k, k)-Anonymization

In this section we describe algorithms for (k, k)-anonymizing a given database D. First, we present in Section V-B.1 algorithms for $(k, 1)$-anonymization. Then, in Section V-B.2, we describe an algorithm for transforming a $(k, 1)$-anonymization into a (k, k)-anonymization.

1) Algorithms for $(k, 1)$-anonymization: Given $D = \{R_1, \ldots, R_n\}$, we may find its optimal $(k, 1)$-anonymization as follows. For each record $R_i \in D$, we look for the subset of $k-1$ records $\{R_{i_1}, \ldots, R_{i_{k-1}}\} \subset D \setminus \{R_i\}$ that minimizes the generalization cost $d(\{R_i, R_{i_1}, \ldots, R_{i_{k-1}}\})$, and then define $\overline{R}_i$ to be the closure of $\{R_i, R_{i_1}, \ldots, R_{i_{k-1}}\}$. As the run time of that algorithm is impractical, $O(n \cdot \binom{n-1}{k-1}) = O(n^k)$, we proceed to describe two approximation algorithms for that problem.

The first one, Algorithm 3, joins each record with the $k-1$ nearest records.

Algorithm 3 $(k, 1)$-anonymization by nearest neighbors

Input: Table D, integer k.
Output: Table $g(D)$ that satisfies $(k, 1)$-anonymity.

1: For all $1 \le i < j \le n$ compute $d_{i,j} = d_{j,i} = d(\{R_i, R_j\})$.
2: **for all** $1 \le i \le n$ **do**
3: Find $k-1$ indices $\{i_1, \ldots, i_{k-1}\} \subset \{1, \ldots, n\} \setminus \{i\}$ that minimize $d_{i,j}$.
4: Define $\overline{R}_i$ to be the closure of $\{R_i, R_{i_1}, \ldots, R_{i_{k-1}}\}$.
5: **end for**

Proposition 5.1: Algorithm 3 produces a table $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ that is a $(k, 1)$-anonymization of D and it approximates optimal $(k, 1)$-anonymization to within a factor of $k - 1$.

While Algorithm 3 offers a guaranteed approximation factor, our second algorithm, Algorithm 4, which constructs the clusters by greedily selecting at each stage the next closest record, performed much better in our experiments. The runtime of both algorithms is $O(kn^2)$.

2) From $(k, 1)$- to (k, k)-anonymization: Let $D = \{R_1, \ldots, R_n\}$ be a database and $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ be any generalization of D. Algorithm 5 further generalizes the records of $g(D)$ until it becomes a $(1, k)$-anonymization of D. By applying this algorithm to a generalization that is already a $(k, 1)$-anonymization, we get a (k, k)-anonymization. (For any $R_i \in D$ and $\overline{R}_j \in g(D)$ we let $R_i + \overline{R}_j$ denote the minimal generalized record that generalizes both R_i and $\overline{R}_j$.)

Algorithm 4 $(k, 1)$-anonymization by expansion

Input: Table D, integer k.
Output: Table $g(D)$ that satisfies $(k, 1)$-anonymity.

1: **for all** $1 \le i \le n$ **do**
2: Set $S_i = \{R_i\}$
3: **while** $|S_i| < k$ **do**
4: Find the record $R_j \notin S_i$ that minimizes $\text{dist}(S_i, R_j) = d(S_i \cup \{R_j\}) - d(S_i)$.
5: Set $S_i = S_i \cup \{R_j\}$.
6: **end while**
7: Define $\overline{R}_i$ to be the closure of S_i.
8: **end for**

Algorithm 5 $(1, k)$-anonymizer

Input: Table $D = \{R_1, \ldots, R_n\}$, generalized table $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$, integer k.
Output: Table $g'(D)$ that generalizes $g(D)$ and satisfies $(1, k)$-anonymity.

1: **for all** $1 \le i \le n$ **do**
2: Let ℓ be the number of records $\overline{R}_j$ that are consistent with R_i.
3: **if** $\ell < k$ **then**
4: Scan all records $\overline{R}_j$ that are not consistent with R_i and find the $k - \ell$ ones that minimize $c(R_i + \overline{R}_j) - c(\overline{R}_j)$.
5: Replace each of those $k-\ell$ records, $\overline{R}_j$, with $R_i + \overline{R}_j$.
6: **end if**
7: **end for**

The runtime of Algorithm 5 is $O(kn^2)$. Consequently, so is the runtime of the coupling of that algorithm with either of the $(k, 1)$-anonymizers, Algorithm 3 or Algorithm 4 (such a coupling is a (k, k)-anonymizer).

C. Global $(1, k)$-Anonymization

Next, we describe Algorithm 6 that transforms a (k, k)-anonymization $g(D)$ of D into a global $(1, k)$-anonymization. The algorithm works as follows: For each $R_i \in D$, it computes the subset P of its set of neighbors Q, consisting of all matches of R_i. Since $g(D)$ is a (k, k)-anonymization of D, then $|Q| \ge k$, but $|P|$ could be less than k. In order to achieve global $(1, k)$-anonymity, we increase $|P|$ so that it becomes at least k. To that end, if $|P| < k$, we select the non-match neighbor $\overline{R}_{j_h}$ of R_i that minimizes the quantity $d_h = c(R_{j_h} + \overline{R}_i) - c(\overline{R}_i)$. Then, we further general the record $\overline{R}_i$ to be consistent also with R_{j_h}. The reader may verify that this update of $\overline{R}_i$ "upgrades" $\overline{R}_{j_h}$ from a neighbor of R_i to a match of R_i. We then keep repeating this procedure until $|P|$ becomes at least k. (It is interesting to note that in almost all of our experiments, one such step was sufficient to increase $|P|$ to become at least k, even if the initial deficiency was greater than 1.)

Algorithm 6 (k,k)- to global $(1,k)$-anonymization

Input: Table $D = \{R_1, \ldots, R_n\}$, generalized table $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ that satisfies (k,k)-anonymity, integer k. (It is assumed that for all $1 \leq i \leq n$, $\overline{R}_i$ is a generalization of R_i.)

Output: Table $g(D)$ that satisfies global $(1,k)$-anonymity

1: **for all** $1 \leq i \leq n$ **do**
2: Set $Q = \{\overline{R}_{j_1}, \ldots, \overline{R}_{j_q}\}$ to be the set of $q \geq k$ neighbors of R_i.
3: Compute P – the subset of Q consisting of all matches of R_i.
4: If $|P| \geq k$, skip to next i.
5: For all $1 \leq h \leq q$ such that $\overline{R}_{j_h} \in Q \setminus P$, compute $d_h = c(R_{j_h} + \overline{R}_i) - c(\overline{R}_i)$.
6: Select the index $1 \leq h \leq q$ where $\overline{R}_{j_h} \in Q \setminus P$, for which d_h is minimal.
7: Set $\overline{R}_i = R_{j_h} + \overline{R}_i$.
8: Return to Step 3.
9: **end for**

Algorithm 6 needs to determine for every edge of the original graph, $(R_i, \overline{R}_j)$, whether it may be completed to a perfect matching in the graph. One way of doing so is by removing the nodes R_i and $\overline{R}_j$ from the graph, and checking whether the remaining graph has a perfect matching. This may be done by invoking the Hopcroft-Karp algorithm, the run time of which is $O(\sqrt{n}m)$, where n is the number of nodes in the graph and m is the number of edges. Since, in the worst case, we need to apply that procedure for every edge, the overall running time of Algorithm 6 is $O(\sqrt{n}m^2)$.

In all of the graphs $V_{D,g(D)}$ that corresponded to our (k,k)-anonymizations of both real and artificial data, the degree of each original record was between k and $2k$. Therefore, $m \leq 2nk$, and, consequently, the running time of Algorithm 6 for our bipartite graphs is $O(n^{2.5}k^2)$. Albeit polynomial in n and k, this runtime may be too large in practice.

VI. Experiment Results

In this section we discuss the experiments that we performed in order to evaluate the new anonymity concepts and our proposed algorithms. We tested all algorithms for k- and (k,k)-anonymization on both artificial and real data.

Artificial data. For a given value of n, we randomly generated tables of n records over a set of six attributes $A_1, \ldots, A_6$. Each of those six attributes consisted of finitely many values that were selected according to the following probability distributions:

A_1 : $\{0.7, 0.3\}$
A_2 : $\{0.3, 0.3, 0.2, 0.2\}$
A_3 : $\{0.25, 0.25, 0.4, 0.1\}$
A_4 : $\{6 \times 0.07, 10 \times 0.04, 9 \times 0.02\}$
A_5 : $\{10 \times 0.1\}$
A_6 : $\{0.05, 0.05, 0.5, 0.3, 0.1\}$

For each of the above attributes, $A = \{a_1, \ldots, a_m\}$, the collection of permissable generalized subsets, $\overline{A}$, is described below. As all of those collections include all singleton subsets, $\{a_i\}$, $1 \leq i \leq m$, as well as the entire set A, we list below only the non-trivial subsets in $\overline{A}$.

$\overline{A}_1$: None (other than$\{a_1\}, \{a_2\}$ and $\{a_1, a_2\}$)
$\overline{A}_2$: $\{a_1, a_2\}, \{a_3, a_4\}$
$\overline{A}_3$: $\{a_1, a_2\}, \{a_3, a_4\}$
$\overline{A}_4$: $\{a_1, \ldots, a_6\}, \{a_7, \ldots, a_{12}\}, \{a_{13}, \ldots, a_{18}\}, \{a_{19}, \ldots, a_{25}\}, \{a_1, \ldots, a_{12}\}, \{a_{13}, \ldots, a_{25}\}$
$\overline{A}_5$: $\{a_1, a_2\}, \{a_3, a_4\}, \{a_6, a_7\}, \{a_8, a_9\}, \{a_1, a_2, a_3, a_4, a_5\}, \{a_6, a_7, a_8, a_9, a_{10}\}$
$\overline{A}_6$: $\{a_1, a_2\}, \{a_4, a_5\}, \{a_3, a_4, a_5\}$

Real-life data. We used two real-life datasets, `Adult` and `Contraceptive Method Choice` (or `CMC`), from the UCI Machine Learning Repository.[1]

`Adult`: This dataset was extracted from the US Census Bureau Data Extraction System. It contains demographic information of a small sample of US population. The public attributes are: age, work-class, education-level, marital-status, occupation, family-relationship, race, sex and native-country. For our experiments we used a subset of the dataset of size $n = 5000$. The collection of permissable generalized subsets in each of the attributes was selected by grouping together values that are semantically close. (For example, the attribute education-level was divided into three groups: high-school, college, and advanced-degrees.) Such generalized databases have more value to the data miner. A complete description of those collections is postponed for the full version of this paper.

`CMC`: This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. Its purpose is to help predicting the contraceptive method choice (no use, long-term methods, or short-term methods) of a woman, based on her demographic and socio-economic characteristics. This dataset has $n = 1500$ records. In the full version of the paper we provide more details on this dataset.

A. Results

The algorithms that we compared were: (a) The two agglomerative algorithms for k-anonymization (Algorithm 1 and its modified version, Algorithm 2); each of them was executed with each of the four distance functions that were described in Section V-A.2. (b) The forest Algorithm, [2]. (c) The two (k,k)-anonymization algorithms that were described in Section V-B (namely, either of Algorithms 3 and 4, coupled with Algorithm 5).

We tested the performance of those algorithms on each of the datasets described above – artificial, adult or CMC (denoted hereinafter as `ART`, `ADT`, and `CMC`, respectively) – and measured the information loss by either the entropy measure (EM) or the LM measure. (In the full version of this paper we used additional measures.)

[1] http://mlearn.ics.uci.edu/MLSummary.html

TABLE I

SUMMARY OF RESULTS

	k	5	10	15	20
ART EM	best k-anon	0.65	0.98	1.13	1.22
	forest	0.89	1.25	1.42	1.51
	(k,k)-anon	0.53	0.83	0.99	1.08
ADT EM	best k-anon	0.66	0.93	1.08	1.18
	forest	1.02	1.45	1.63	1.73
	(k,k)-anon	0.50	0.75	0.90	1.00
CMC EM	best k-anon	0.67	0.95	1.08	1.20
	forest	0.99	1.31	1.46	1.53
	(k,k)-anon	0.54	0.80	0.98	1.10
ART LM	best k-anon	0.12	0.19	0.23	0.25
	forest	0.15	0.24	0.28	0.31
	(k,k)-anon	0.10	0.16	0.19	0.22
ADT LM	best k-anon	0.14	0.20	0.24	0.26
	forest	0.22	0.37	0.46	0.53
	(k,k)-anon	0.09	0.13	0.16	0.18
CMC LM	best k-anon	0.14	0.21	0.25	0.28
	forest	0.19	0.31	0.40	0.44
	(k,k)-anon	0.11	0.17	0.20	0.23

Our results are summarized in Table I. They are partitioned into six sets of experiments according to the choice of dataset and measure. In each set, the first row ("best k-anon") shows the results of the agglomerative k-anonymization algorithm that minimized the sum of information loss over four experiments with $k = 5, 10, 15, 20$. The second row shows the result of the forest algorithm, and the third one shows the result of the better (k,k)-anonymization. Figures 2 and 3 illustrate the results for the adult database. (The graphs for the other two databases are similar.)

The main conclusions of our experimental results are:

• All of the suggested agglomerative k-anonymity algorithms yield better anonymizations than the forest algorithm; information loss is reduced by 20%–50%.

• The improvement offered by (k,k)-anonymity over the best k-anonymity algorithm, ranges between 10% and 30%.

Additional conclusions that arise from our experiments are as follows (more details will be provided in the full version of this paper):

• Among the different variants of the k-anonymity agglomerative algorithms, the two distance functions that consistently bring the best results are (10) and (11).

• In all of the experiments, the coupling of Algorithms 4 and 5 produced better (k,k)-anonymizations than the coupling of Algorithms 3 and 5.

• The corrections made in the modified agglomerative algorithm usually reduce the information loss, as expected. However, those improvements are negligible for the two distance functions mentioned above, because these functions were designed so that the resulting clusters have the required size, thus leaving only little room for improvement.

Another interesting finding from Table I is that the average information loss per entry remains roughly the same for each of our algorithms, regardless of the dataset. For example, the best k-anonymization algorithm loses about 0.66 bits of information per entry, and about 0.13 LM-"information units" per entry, in all datasets.

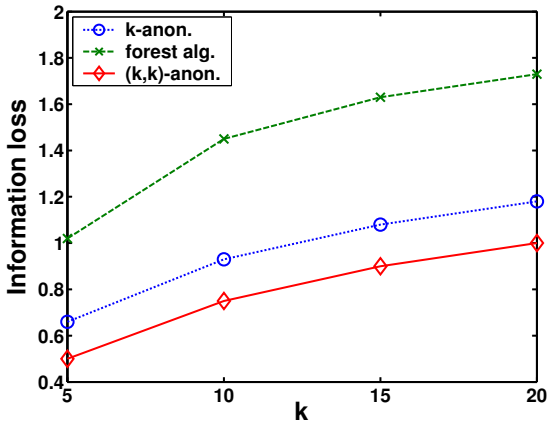

Fig. 2. Comparison of algorithms by the entropy measure

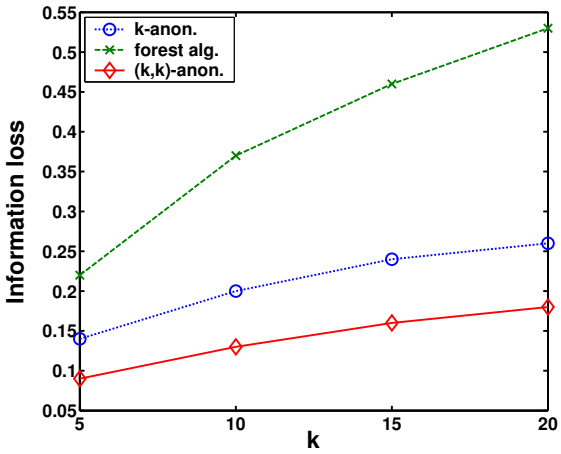

Fig. 3. Comparison of algorithms by the LM measure

VII. CONCLUSIONS

In this paper we proposed new notions of k-type anonymizations. Our goal has been to ensure anonymization of a dataset while minimizing the amount of information lost during the anonymization process. The anonymity concepts we defined are called $(1,k)$-, $(k,1)$-, (k,k)-, and global $(1,k)$-anonymizations, all of which are relaxations of the original k-anonymity notion, hence they offer solutions with higher utility. As $(1,k)$- and $(k,1)$-anonymity were exemplified to be weak, we proposed (k,k)-anonymity and global $(1,k)$-anonymity as more secure notions. (k,k)-anonymizations are secure if we assume that the adversary has access to a limited amount of records in the dataset, but can be insecure against a powerful adversary that has full knowledge of all public records. On the other hand, global $(1,k)$-anonymity is as secure as k-anonymity, even against such powerful adversaries.

We described algorithms for k-anonymity and for the new k-type anonymity notions. Our experiments showed that our new agglomerative k-anonymity algorithms perform in practice better than algorithms previously proposed in the literature for the same problem (for the local recoding model). Also,

we verified that (k, k)-anonymization yields indeed solutions that have smaller amount of information loss than solutions obtained by k-anonymization.

Many interesting problems remain for future work. One is to find more scalable algorithms or algorithms with better approximation guarantees. Experimentally, we would like to explore the relation between (k, k)-anonymity and global $(1, k)$-anonymity. For instance, for real-life datasets, it might be true that (k, k)-anonymization (or perhaps a $((1+\varepsilon)k, (1+\varepsilon)k)$-anonymization for a suitably chosen ε) yields solutions that satisfy also global $(1, k)$-anonymity.

REFERENCES

[1] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu. Achieving anonymity via clustering. In *PODS*, 2006.

[2] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *ICDT*, 2005.

[3] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for k-anonymity. *Journal of Privacy Technology*, 2005.

[4] G. Aggarwal, N. Mishra, and B. Pinkas. Secure computation of the kth-ranked element. In *Eurocrypt*, 2004.

[5] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMOD*, 2000.

[6] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *ICDE*, 2005.

[7] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *PODS*, 2005.

[8] T. Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15:429–444, 1977.

[9] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, 2003.

[10] A. Gionis and T. Tassa. k-Anonymization with minimal loss of information. In *ESA*, 2007.

[11] V. Iyengar. Transforming data to satisfy privacy constraints. In *KDD*, 2002.

[12] D. Kifer and J. Gehrke. Injecting utility into anonymized datasets. In *SIGMOD*, 2006.

[13] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Auditing boolean attributes. *JCSS*, 6:244–253, 2003.

[14] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Incognito: efficient full-domain k-anonymity. In *SIGMOD*, 2005.

[15] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *ICDE*, 2006.

[16] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *PODS*, 2004.

[17] M. Nergiz and C. Clifton. Thoughts on k-Anonymization. In *ICDE Workshops*, 2006.

[18] P. Samarati. Protecting respondent's privacy in microdata release. *TKDE*, 13:1010–1027, 2001.

[19] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In *PODS*, 1998.

[20] J. Vaidya, Y. M. Zhu, and C. W. Clifton. *Privacy Preserving Data Mining*. Springer-Verlag, 2005.

[21] X. Xiao and Y. Tao. Anatomy: Simple and Effective Privacy Preservation. In *VLDB*, 2006.

[22] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu. Utility-based anonymization using local recoding. In *KDD*, 2006.