TRANSACTIONS ON DATA PRIVACY 5 (2012) 189–222

k-Concealment: An Alternative Model of k-Type Anonymity

Tamir Tassa[*], **Arnon Mazza**[**], **Aristides Gionis**[***]

[*] Department of Mathematics and Computer Science, The Open University, Ra'anana, Israel.

(Corresponding author. Email: tamirta@openu.ac.il)

[**] Department of Mathematics and Computer Science, The Open University, Ra'anana, Israel.

[***] Yahoo! Research, Barcelona, Catalunya, Spain.

Abstract. We introduce a new model of k-type anonymity, called k-concealment, as an alternative to the well-known model of k-anonymity. This new model achieves similar privacy goals as k-anonymity: While in k-anonymity one generalizes the table records so that each one of them becomes equal to at least $k-1$ other records, when projected on the subset of quasi-identifiers, k-concealment proposes to generalize the table records so that each one of them becomes computationally - indistinguishable from at least $k-1$ others. As the new model extends that of k-anonymity, it offers higher utility. To motivate the new model and to lay the ground for its introduction, we first present three other models, called $(1,k)$-, $(k,1)$- and (k,k)-anonymity which also extend k-anonymity. We characterize the interrelation between the four models and propose algorithms for anonymizing data according to them. Since k-anonymity, on its own, is insecure, as it may allow adversaries to learn the sensitive information of some individuals, it must be enhanced by a security measure such as p-sensitivity or ℓ-diversity. We show how also k-concealment can be enhanced by such measures. We demonstrate the usefulness of our models and algorithms through extensive experiments.

1 Introduction

A vast amount of information is collected on a regular basis about individuals by various organizations. In today's global network of organizational connections, there is a growing demand to disseminate and share this information due to various academic, commercial and other benefits. As the records of data frequently include sensitive information that could violate the privacy of the corresponding individuals, it is necessary to preprocess the data prior to its publication in order to limit the disclosure of sensitive data. Such preprocessing operations usually involve data distortion. The challenge is then to preprocess the data so that some privacy measure is met, on one hand, while the utility of the data is preserved, on the other hand.

Privacy Preserving Data Publishing (PPDP) is an evolving research field that is targeted at developing techniques to enable publishing data so that privacy is preserved while data

[*] Aristides Gionis was partially supported by the Torres Quevedo Program of the Spanish Ministry of Science and Innovation, co-funded by the European Social Fund, and by the Spanish Centre for the Development of Industrial Technology under the CENIT program, project CEN-20101037, "Social Media" (www.cenitsocialmedia.es).

distortion is minimized [16]. It is closely related to Privacy Preserving Data Mining (PPDM) [4, 43]. The latter term is usually reserved to settings in which the data mining tasks are known when designing the corresponding privacy-preserving algorithms; PPDP, on the other hand, usually refers to settings in which the purposes of the data release are unknown and it is needed to anonymize the data using general purpose utility measures that are not targeted at specific data mining goals.

PPDP assumes that the candidate table to be published includes four types of attributes [9]: identifiers — attributes that uniquely identify an individual (e.g. `S.S.N.`); quasi-identifiers — attributes like `occupation`, `age`, or `gender`, that do not offer unique identification but their combination might yield unique identification; non-identifiers — non-sensitive attributes that are not quasi-identifiers, in the sense that an adversary is unlikely to get hold of them; and private attributes — personal attributes of sensitive nature, such as health condition. A usual practice in PPDP is to remove the identifiers and to generalize the quasi-identifiers in order to limit the disclosure of private data.

One of the most well-studied models of PPDP is k-anonymization [2, 6, 28, 33, 38, 39]. In that model, the quasi-identifiers of the table records are generalized until each record becomes identical to at least $k-1$ other records, when projected on the subset of the quasi-identifiers. A general purpose *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.

1.1 Motivation

In this study we assume that the adversary knows the public data of all individuals in the population, and that he knows the exact subset of the population that is represented in the table. Such an adversary may extract from a publicly accessible database the subset of records which relate to the individuals that appear in the released anonymized table. The former table includes the quasi-identifiers of all records, as well as identifying information. The latter table includes the generalized quasi-identifiers as well as sensitive information. The adversarial task is to find the correct mapping between the records of the two tables. k-Anonymity ensures that each record in the first table can be linked to no less than k records in the anonymized table. To achieve that, it requires that the anonymized table will consist of clusters of size at least k, where all records in the same cluster have the same generalized quasi-identifiers. Consequently, it is impossible to distinguish between the records of a given cluster in the information-theoretic sense, whence, the adversary cannot tell which of the k (or more) records is the one for which he is looking.

We argue that the k-anonymity condition is unnecessarily rigid and leads to excessive generalization of the quasi-identifiers, and, consequently, to unnecessary information losses. We propose here an alternative model of *k-concealment*. It extends the model of k-anonymity in the sense that every table that is k-anonymized satisfies also k-concealment, but the converse does not necessarily hold. In k-concealment we replace the information-theoretic hiding with computational indistinguishability. Namely, there are no more clusters of identical records; instead, every record can be matched with at least k generalized records and even the potent adversary that was described above would not be able to distinguish between them assuming that he is polynomially-bounded (i.e., he may perform only polynomial-time computations). Hence, from a practical point of view, k-concealment is as secure as k-anonymity, just as much as modern ciphers are accepted as secure alternatives to information-theoretic perfectly secure ciphers. The advantage that is offered by this new model is enhanced utility: k-Concealment may be achieved with less generalization than that which is required by k-anonymity.

Example. Consider the basic table in Table 1(a), having quasi-identifiers `age` and `zipcode` and the sensitive attribute `disease`. Table 1(b) is a corresponding 2-anonymization. It consists of two clusters of records that have the same generalized quasi-identifiers — the first three records and the last two. It is impossible to distinguish between the records in the same cluster, because they are identical. Table 1(c), on the other hand, is a 2-concealment of Table 1(a). An adversary who knows the quasi-identifiers of all records in Table 1(a) cannot link any such record with less than two generalized records in Table 1(c). For example, based on the two quasi-identifiers `age` and `zipcode`, the adversary cannot tell whether Alice's record in Table 1(c) is the first or the third one. Those two records are not identical; thus, in theory, the adversary may be able to deduce that one of those records is more likely to be Alice's than the other. We claim, and explain later on, that it is computationally hard to do so. Hence, for polynomially-bounded adversaries, these two candidate records are equally likely to be Alice's generalized record. Finally, as can be seen, Table 1(c) involves less data distortion than Table 1(b) (the entries in Table 1(c) that are more specific are marked).

name	age	zipcode	disease
Alice	30	10055	Measles
Bob	21	10055	Flu
Carol	21	10023	Angina
David	55	10165	Flu
Eve	47	10224	Diabetes

(a) The original table

age	zipcode	disease
21-30	100**	Measles
21-30	100**	Flu
21-30	100**	Angina
47-55	10***	Flu
47-55	10***	Diabetes

(b) 2-Anonymization

age	zipcode	disease
21-30	**10055**	Measles
21	100**	Flu
21-30	100**	Angina
47-55	10***	Flu
47-55	10***	Diabetes

(c) 2-Concealment

Table 1: A table and corresponding anonymizations

1.2 Is k-anonymity still relevant?

Several studies have pointed out weaknesses of the k-anonymity model and suggested more secure measures such as ℓ-diversity [32], t-closeness [30], or p-sensitivity [42]. The main weakness of k-anonymity is that it does not guarantee sufficient diversity in the private attribute in each equivalence class of indistinguishable records. Namely, even though it guarantees that every record in the anonymized table is indistinguishable from at least

$k-1$ others, it is possible that all of those records, that agree in their generalized quasi-identifiers, are also equal in their private value. Therefore, an adversary who is capable of locating his target individual in that block of records, will be able to infer the private value of that individual. Machanavajjhala et al. [32] proposed the security measure of ℓ-diversity. They suggested that the private attribute in each block will have at least ℓ "well represented" values. They offered two interpretations of that measure. In one interpretation, the entropy of the values in that attribute in every block should be at least $\log \ell$, for some predetermined value of the parameter ℓ. The other interpretation is that of recursive (c, ℓ)-diversity (see [32] for its definition). According to a simpler interpretation of ℓ-diversity [47, 48], a block is ℓ-diverse if the relative frequency of each of the private values within each block is at most $1/\ell$.

It is important to understand that those notions do not and can not replace k-anonymity. They offer essential *enhancements* to k-anonymity in the sense that one must require them *in addition* to k-anonymity. In accord with this, Truta et al. [42] proposed algorithms that generate tables that are both k-anonymous and p-sensitive, and Wong et al. [47] considered the conjunction of k-anonymity with the last interpretation of ℓ-diversity (they call this conjunction of conditions $(1/\ell, k)$-anonymity).

In order to clarify that point, let us consider the measure of ℓ-diversity. The diversity of a table is bounded from above by the number of possible private values (equality holds if and only if the distribution of the private values is uniform). The diversity of any anonymization of the table is bounded from above by the the diversity of the entire table (equality holds if and only if the distribution in each block equals the global distribution). Therefore, if the table has a private attribute with a small number of possible values, all of its anonymizations will respect ℓ-diversity with ℓ that does not exceed this number. For example, in the case of a binary private attribute, one can aim at achieving ℓ-diverse anonymizations with $\ell \leq 2$ only. In such a case, if one imposes only ℓ-diversity, the blocks of indistinguishable records could be of size 2. Such small blocks do not provide enough privacy for the individuals in them, because if an adversary may be able to learn the private value of one of those individuals, he may infer that of the other one as well. If, on the other hand, we demand that such ℓ-diverse anonymizations are also k-anonymous, for a suitable selection of k, then the adversary would have to find out the private values of at least $k/2$ individuals before he would be able to infer the private value of his target individual. Hence, k-anonymity is still a vital notion that serves as a basis to ℓ-diversity.

Another reason why ℓ-diversity cannot stand alone and must be accompanied by k-anonymity is that it is defined only through the distribution of the sensitive values in each block; as a consequence, it is vulnerable to minimality attacks [46]. A useful tool in combating such attacks is to apply k-anonymization with an information loss measure that considers only the quasi-identifiers (and not the sensitive attributes), and only then transfer the k-anonymized table into one that also respects ℓ-diversity (for a more detailed explanation, see [46]).

1.3 Outline and contributions

In this paper we offer the following contributions:

- We introduce new models of k-type anonymizations (namely, $(k, 1)$-, $(1, k)$-, and (k, k)-anonymity and k-concealment) that lead to anonymized tables with higher utility. We characterize the relations among the new anonymity models and the original model of k-anonymity.

- We show that the model of k-concealment offers a comparable level of security to that of k-anonymity.
- We describe algorithms to achieve (k,k)-anonymity and k-concealment.
- We show how k-concealment may be enhanced using p-sensitivity or ℓ-diversity.
- We demonstrate the usefulness of our definitions and our proposed algorithms through experimental evaluation on real and synthetic datasets.

The rest of the paper is organized as follows. In Section 2 we discuss related work. In Section 3 we formally define the basic concepts, and in Section 4 we introduce the new models of k-type anonymity and discuss their interrelations. Section 5 is devoted to discussing the security of those models and showing that k-concealment is essentially as secure as k-anonymity. In Section 6 we propose algorithms for (k,k)-anonymization and k-concealment. Since we assume that the adversary knows the anonymization algorithms, it is necessary to randomize them; this is done in Section 7. In Section 8 we discuss enhancements of our algorithms so that they issue k-concealed tables that respect also p-sensitivity or ℓ-diversity. In Section 9 we describe the experiments that we executed for testing the performance of those algorithms. Finally, we conclude our discussion in Section 10.

A preliminary version of this work [19] included the definitions of the new models and the algorithms to achieve (k,k)-anonymity. The current work extends [19] by proposing new algorithms for k-concealment and providing a thorough security analysis of that notion, under stronger adversarial assumptions. In addition, we propose here a randomized version of our algorithm as a countermeasure against minimality attacks. We discuss enhancements of our models to support also p-sensitivity and ℓ-diversity. Finally, our experimentation here significantly extends that which we reported in [19].

2 Related work

The objective of protecting the privacy of individuals represented in databases was formulated by Dalenius [12] in 1977. Since then, many approaches have been suggested for finding the right balance between data hiding and data disclosure. Such approaches include query auditing [27], output perturbation [7], secure multi-party computation [3], and data sanitization [4, 13].

One such approach, originally proposed by Samarati and Sweeney [37, 38, 39], is k-anonymization. Meyerson and Williams [33] 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 a runtime of $O(n^{2k})$ and an approximation ratio of $O(k \log k)$, and another with a fully polynomial runtime and an approximation ratio of $O(k \log n)$. Aggarwal et al. [2] 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] is defined as a *tree measure* and it is a generalization of the function considered by Meyerson and Williams [33]. In [20], three *entropy-based* functions were 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 $O(\log k)$-approximation algorithm was presented in [20] for the problem of optimal k-anonymity with respect to two of the entropy-based measures, as well as for the tree measure. More efficient $O(\log k)$-approximation algorithms were described in [35], for the case of suppressions only, and in [24], for the general case.

Other information-loss measures were used in previous studies. The LM measure [22, 34] (which we define later on in Section 4.1) is a more precise version of the tree measure of [2]. The CM measure [22] 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 [20], 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.

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.

In a similar line to our present work, the works of Kifer and Gehrke [26] and Xiao and Tao [48] aim at improving the utility of the anonymized data. Kifer and Gehrke [26] suggested 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 [48] proposed 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.

3 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 a private attribute, A_{r+1}. Each of the attributes consists of several possible values:

$$A_j = \{a_{j,\ell} : 1 \le \ell \le m_j\},\ 1 \le j \le r+1\,.$$

For example, if A_j is the attribute `gender` then $A_j = \{\mathtt{M},\mathtt{F}\}$, while if A_j is the attribute `age`, then it is a bounded natural number. (Note that we use A_j to denote the attribute as well as the domain in which it take values.)

Hereinafter, D denotes the projection of the database on the set of r public attributes and the records of D are denoted R_i, $1 \le i \le n$; namely, $R_i \in A_1 \times \cdots \times A_r$. We denote the jth 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 1. Let A_j, $1 \le j \le r$, be finite sets and let $\overline{A}_j \subseteq \mathcal{P}(A_j)$ be a collection of subsets of A_j. A record $\overline{R} \in \overline{A}_1 \times \cdots \times \overline{A}_r$ is a generalization of the record $R \in A_1 \times \cdots \times A_r$ (or, alternatively, it is consistent with R) if $R(j) \in \overline{R}(j)$ for all $1 \le j \le r$.

If $D = \{R_1, \ldots, R_n\}$ is a table of records in $A_1 \times \cdots \times A_r$, then $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ is a generalization of D if for all $1 \le i \le n$, $\overline{R}_i$ is a generalization of R_i.[1]

[1] Hereinafter D and $g(D)$ are multisets, in the sense that they may include differently-indexed records that are equal.

As an example, consider a database D with two quasi-identifiers, `age` (A_1) and `zipcode` (A_2). Then the generalized record $\overline{R} = (\{30, \ldots, 39\}, \{98000, \ldots, 98099\})$ is a generalization of the record $R = (34, 98003)$.

We shall assume that the collection of subsets $\overline{A}_j$, $1 \leq j \leq r$, that may generalize the values of the attribute A_j are proper, in the following sense.

Definition 2. Given an attribute $A = \{a_1, \ldots, a_m\}$, a corresponding collection of subsets $\overline{A}$ is called proper if it includes all singleton subsets $\{a_i\}$, $1 \leq i \leq m$, it also includes the entire set A, and it is laminar in the sense that $B_1 \cap B_2 \in \{\emptyset, B_1, B_2\}$ for all $B_1, B_2 \in \overline{A}$.

The first part of Definition 2 implies that with such collections of subsets, we may leave each value of that attribute unaltered (no generalization), or replace it by the entire set of values for that attribute (total generalization, or suppression). The second part of the definition that restricts the subset collection $\overline{A}$ to be *laminar* implies that $\overline{A}$ forms a hierarchy (see [20, Lemma 3.3].)

As a final note, we distinguish between three main models of generalization:

- In *(single-dimensional) global recoding*, e.g. [6, 22, 28, 47], each collection of subsets $\overline{A}_j$ is a partition of the set A_j (in the sense that $\overline{A}_j$ includes *disjoint* sets whose union equals A_j). In such cases, every entry in the jth column of the database is mapped to the unique subset in $\overline{A}_j$ that contains it. As a consequence, every single value $a \in A_j$ is always generalized in the same manner.

- In *local recoding*, e.g. [19, 20, 33, 35, 47], the collection of subsets $\overline{A}_j$ covers the set A_j but it is not a partition (namely, the subsets in the collection may intersect). In such cases, each entry in the table's jth column is generalized independently to one of the subsets in $\overline{A}_j$ which includes it. Hence, if the age 34, for example, appears in the table in several records, it may be left unchanged in some, or generalized to 30 - 39, or totally suppressed in other records.

- The third model is an intermediate one and it is called *multi-dimensional global recoding*, e.g. [29]. In that model, like in local recoding, the collection of subsets $\overline{A}_j$ is a cover of the set A_j (namely, each value of A_j may be contained in more than one subset in $\overline{A}_j$). However, it is a global recoding in the sense that there exists a global mapping function $g : A_1 \times \cdots \times A_r \to \overline{A}_1 \times \cdots \times \overline{A}_r$ such that $\overline{R}_i = g(R_i)$ for all $1 \leq i \leq n$.

4 Alternative models of k-type anonymity

We begin this section by reviewing the notion of k-anonymity as it is used in the recent literature [2, 6, 20, 28, 33]. 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, whence they allow greater utility.

4.1 Overview of k-anonymization

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 $\Pi(D, g(D))$ denote the amount of information that is lost by replacing a database D with a corresponding generalization $g(D)$. The measure of loss of information Π took several forms in previous studies; the reader is referred to [19] for an overview of some of the commonly used measures. Most of those measures are additive in the sense that they associate an information loss with each generalized record and then the overall information loss in $g(D)$ is the sum of information losses of all generalized records in $g(D)$. Given a generalized record $\overline{R} \in \overline{A}_1 \times \cdots \times \overline{A}_r$, we let $c(\overline{R}) = c_\Pi(\overline{R})$ denote the cost by which $\overline{R}$ is penalized according to the chosen measure Π of information loss.

For the sake of illustration, we recall here the definition of the commonly used Loss Metric (LM) measure [22, 34]. If $\overline{R}$ is a generalized record in $\overline{A}_1 \times \cdots \times \overline{A}_r$, then the LM measure associates with it the following cost,

$$c(\overline{R}) = \frac{1}{r} \cdot \sum_{j=1}^{r} \frac{|\overline{R}(j)| - 1}{|A_j| - 1} . \tag{1}$$

Namely, it is an average cost over all r attributes, where the cost in the jth attribute ranges between 0 (no generalization at all) to 1 (total suppression). If $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ is a generalization of $D = \{R_1, \ldots, R_n\}$, then the overall generalization cost by LM is

$$\Pi(D, g(D)) := \frac{1}{n} \cdot \sum_{i=1}^{n} c(\overline{R}_i) . \tag{2}$$

As mentioned earlier, we use in our experiments the LM measure as well as the basic entropy measure (EM) of [20]; the reader is referred to [20] for a definition of the latter measure.

Definition 3. Let $S \subset A_1 \times \cdots \times A_r$ be a set of records. Its closure is the minimal generalized record in $\overline{A}_1 \times \cdots \times \overline{A}_r$ that is consistent with all records in S. The generalization cost of S is then defined as $d(S) = c(\overline{S})$.

4.2 k-type anonymizations

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 1.

Definition 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 called a $(1, k)$-anonymization of D if each record in D is consistent with at least k records in $g(D)$.
- $g(D)$ is called a $(k, 1)$-anonymization of D if each record in $g(D)$ is consistent with at least k records in D.
- $g(D)$ is called 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 k-, $(1, k)$-, $(k, 1)$- and (k, k)-anonymizations of the database D, respectively.

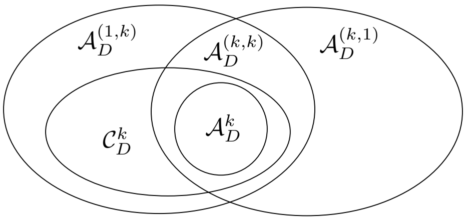

Figure 1: Interrelations between the five classes of k-type anonymizations.

In order to understand the motivation behind those definitions, we look at things from the perspective of the adversary. A typical adversarial attack aims at revealing sensitive information on a specific target individual. In such an attack, the adversary knows a record $R \in D$ and he tries to locate the corresponding generalized record $\overline{R} \in g(D)$. Alternatively, the adversary could be interested in re-identifying any entity in the released data, for instance to find possible victims to blackmail. Such an attack works in the opposite direction: Focusing on a generalized record $\overline{R} \in g(D)$, the adversary tries to infer its correct preimage $R \in D$. (This kind of attack was at the basis of the well-known August 2006 AOL crisis, see [5].) The notion of $(1,k)$-anonymity aims at protecting against the first attack; the notion of $(k,1)$-anonymity aims at protecting against the second one; (k,k)-anonymity considers both attacks.

The notion of $(k,1)$-anonymity was already defined in [45] under the name k-ambiguity. A similar security notion appeared in [39]: an anonymized table adheres to the k-map protection model if every record in it is consistent with at least k entities in the underlying population (and not just in the original table, as in k-ambiguity and its equivalent $(k,1)$-anonymity).

Proposition 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. Then the relation between these collections is as depicted in Figure 1; i.e.,*

$$\mathcal{A}_D^k \subsetneqq \mathcal{A}_D^{(k,k)} \subsetneqq \mathcal{A}_D^{(1,k)}, \mathcal{A}_D^{(k,1)}, \tag{3}$$

and

$$\mathcal{A}_D^{(1,k)} \setminus \mathcal{A}_D^{(k,1)} \neq \emptyset, \quad \mathcal{A}_D^{(k,1)} \setminus \mathcal{A}_D^{(1,k)} \neq \emptyset. \tag{4}$$

(The proof of the propositions in this section are given in Appendix A.)

Our anonymity definitions can also be understood via graph terminology, as follows:[2] 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 our final and main notion:

Definition 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

[2] All graph terminology that we use herein is defined and discussed in any basic textbook on graph theory, e.g. [8].

edge in $V_{D,g(D)}$ and there exists a perfect matching in $V_{D,g(D)}$ that includes that edge. If all records $R \in D$ have at least k matches in $g(D)$, then $g(D)$ is called a k-concealment of D.

Namely, k-concealment is a stronger version of $(1,k)$-anonymity: While $(1,k)$-anonymity required each record (or node) $R \in D$ to have at least k adjacent edges in $V_{D,g(D)}$, k-concealment demands that each record $R \in D$ will have at least k adjacent *matches*. We could define similar stronger versions of $(k,1)$-anonymity and (k,k)-anonymity. Herein we choose to focus on k-concealment as the stronger version of $(1,k)$-anonymity since the adversarial attack that motivates $(1,k)$-anonymity is the more interesting one (see the discussion after Definition 4). In addition, the algorithms that we present in this paper for the notion of k-concealment can be easily modified so that they apply to the concealment versions of $(k,1)$-anonymity and (k,k)-anonymity.

The relation between the new anonymization class, denoted $\mathcal{C}_D^k$, and the previous ones is given in the following proposition.

Proposition 7. *Let $\mathcal{C}_D^k$ denote the collection of all k-concealments 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 $\mathcal{C}_D^k$, is as depicted in Figure 1.*

We would like to note that our definitions of the alternative models of k-type anonymity are relevant only in the case of local recoding. It is easy to see that in case of either single- or multi-dimensional global recoding all those models coincide with the standard k-anonymity (namely, all domains in Figure 1 collapse into one domain).

5 The security of the new k-type notions

In Section 5.1 we discuss the security of the three basic notions of $(1,k)$-, $(k,1)$- and (k,k)-anonymity. We explain why they do not provide comparable privacy to that offered by k-anonymity. This discussion provides the motivation for the central notion which we introduce in this study — k-concealment. Then, in Section 5.2 we discuss the security of that notion. We show there that k-concealment offers comparable security as k-anonymity (but with lower information losses).

5.1 The insecurity of $(1,k)$-, $(k,1)$- and (k,k)-anonymity

Here we discuss the security of the three basic notions of $(1,k)$-, $(k,1)$- and (k,k)-anonymity. We show that they do not provide the same level of security as k-anonymity. The purpose of this discussion is to motivate the definition of k-concealment, which, as we show later on, provides a comparable level of security as k-anonymity (with higher utility, as shown by experimentation).

Consider a database D and a corresponding $(k,1)$-anonymization $g(D)$. Each record in $g(D)$ is consistent with at least k records in D. However, it is possible that some records $R \in D$ are consistent with only one record in $g(D)$, as illustrated in Figure 2. (It describes a $(2,1)$-anonymization but the first record in D is consistent only with one record in $g(D)$.). If the adversary happens to target an individual whose quasi-identifier record in D has only one generalized record in the released table that is consistent with it, then that adversary may infer, with certainty, what is the generalized record that corresponds to his target individual and, consequently, find the corresponding private attribute of that record.

Next, let $g(D)$ be a $(1,k)$-anonymization of D. It is true that every record in D is consistent with at least k records in $g(D)$, whence such anonymizations seem to satisfy our privacy

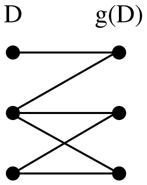

Figure 2: An example of a breach of security in $(k, 1)$-anonymity

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 $1 \le i \le n-k$ and $\overline{R}_i = \overline{R}^*$ for all $n-k+1 \le i \le n$ (the corresponding bipartite graph for $n = 5$ and $k = 2$ is shown in Figure 3). 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.

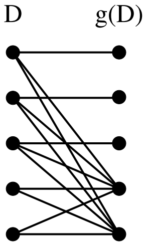

Figure 3: An example of a breach of security in $(1, k)$-anonymity

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. However, since we assumed that the adversary knows the exact subset of the population that is represented in the table, and that he knows the public data of all of them, he may construct the original public table D. Since the anonymized table $g(D)$ is published, he may construct the bipartite graph $V_{D,g(D)}$, where an edge connects a record $R_i \in D$ to a generalized record $\overline{R}_j \in g(D)$ if and only if the latter generalizes the former. The (k, k)-anonymity guarantees that every node in that graph has a degree of at least k. However, while this lower bound on the degree may be sufficient in case the adversary knows only part of D, it is insufficient under our strong adversarial assumption. Since the adversary knows the entire graph, he may test each edge in the graph and check whether it may be completed into a perfect matching in the graph. If it cannot, then it does not represent a possible link between an original record and a generalized one. By testing the edges of the graph and removing edges that are not matches, in the sense of Definition 6, the adversary may end up with nodes in D that have less than k matches, whence he may obtain more linkage information than what we were expecting.

This discussion motivates our definition of k-concealment, which, as we proceed to show next, is secure against the potent adversary that we consider. However, before moving on to discuss the security of k-concealment, we would like to point out that from practical point of view, assuming that the adversary knows the exact subset of the population that is represented in the table is a very strong one. In fact, the data holder can simply eliminate at random a small subset of records in order to make it impossible for such an adversary to construct the graph $V_{D,g(D)}$ and apply the above described attack on (k,k)-anonymous tables. Hence, we believe that in practice the model of (k,k)-anonymity offers a comparable security to that offered by k-anonymity. Having said that, we proceed to discuss the security of our main model — k-concealment.

5.2 Security of k-concealment

In k-concealment, every $R_i \in D$ has at least k matches in $g(D)$. Hence, just like in k-anonymity, even the strongest adversary, as we described earlier, who targets a particular record $R_i \in D$, can not narrow down the number of suspect generalized records in $g(D)$ to less than k.

However, while in k-anonymity the k suspect records in $g(D)$ are identical, whence each of them is equally likely to be the real generalization of R_i, the same is not true for k-concealment. Assume that the adversary, who can construct $V_{D,g(D)}$, wishes to "attack" a given record in D, say R_1. Let $\{\overline{R}_{i_1}, \ldots, \overline{R}_{i_t}\}$ be the set of matches of R_1 in $g(D)$. The adversary may count, for each $1 \leq j \leq t$, the number of perfect matchings in $V_{D,g(D)}$ that include the edge $(R_1, \overline{R}_{i_j})$. Denoting those counts by m_j, he may deduce that the probability that $\overline{R}_{i_j}$ is the true match of R_1 is $p_j := \frac{m_j}{M}$, where $M = \sum_{j=1}^{t} m_j$ is the total number of perfect matchings in $V_{D,g(D)}$. As the probabilities $p_1, \ldots, p_t$ are not necessarily equal, that adversary may gain an advantage which could have not been gained in the k-anonymity model.

However, that attack is infeasible since counting the number of perfect matchings in a bipartite graph is equivalent to computing the permanent of a $\{0,1\}$-matrix[3]. That problem has a rich history in the study of computational complexity. All known algorithms for computing the permanent over the integers, or over any finite field of odd characteristic, are exponential. The best known algorithm runs in time $O(n^2 2^n)$ [36]. In 1979, Valiant proved that the problem is in #P-complete [44]. A decade later, Toda [41] demonstrated the surprising power of #P; Toda's theorem, combined with Valiant's result, implies that the permanent is hard for the entire polynomial-time hierarchy.

The permanent is hard not only in the worst case, but also in the average case. Lipton [31] has shown that the permanent has the random self-reducibility property. The important consequence of that property is that the permanent is hard also on the average. The line of research initiated by Lipton, that connects the worst case and average case complexities of the permanent was pursued in several studies [14, 17, 18]. The best result in that direction is due to Cai et al. [10]: They proved that if there exists a polynomial time algorithm (even a probabilistic one) that computes the permanent of a matrix of order n for any inverse polynomial fraction of all matrices of order n, then there is a probabilistic polynomial time algorithm that computes the permanent for every matrix.

Because of its computational difficulty, there has been much research on polynomial time approximation algorithms for the permanent. The best currently available approximation is a fully polynomial randomized approximation scheme [23] that provides an arbitrarily

[3] The permanent of a square matrix $A = (a_{i,j})_{1 \leq i,j \leq n}$ is defined as perm(A)= $\sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$ where S_n is the group of all permutations of $\{1, 2, \ldots, n\}$.

close approximation. However, the runtime of that algorithm is $\tilde{O}(n^{10})$, thus it is infeasible even for very modest databases with n in the thousands.

To summarize, even an adversary who can construct the entire graph $V_{D,g(D)}$ cannot locate his target individual within less than k suspect records in the anonymized table. Furthermore, he cannot even distinguish between those suspect records, since, in order to do that, he must solve at least k #P-complete problems, or at least use approximation algorithms which currently are also infeasible. Given all of the above, we conclude that k-concealment is essentially as secure as k-anonymity.

6 Algorithms

Algorithms for k-anonymization may be separated to approximation algorithms and heuristic algorithms. Approximation algorithms issue, for a given table D, a k-anonymization $g(D)$ with an approximation ratio guarantee ρ; namely, if $g_o(D)$ is an optimal k-anonymization of D then it is guaranteed that $\Pi(D, g(D)) \leq \rho \cdot \Pi(D, g_o(D))$. The *forest algorithm*, by Aggarwal et al. [2], offers an approximation ratio guarantee of $3k - 3$ with respect to a tree measure of information loss. (A better approximation algorithm with an approximation ratio of $O(\log k)$ was presented in [35] but it is limited to generalization by suppression only.) In practice, however, better results may be obtained by heuristic algorithms. The two algorithms that appear to be the best ones are the agglomerative algorithm [19] and the sequential algorithm [21]. As demonstrated in [21], the two algorithms offer very similar results in terms of utility.

In this section we describe algorithms for (k, k)–anonymization and k-concealment, and compare their performance to that of the above mentioned algorithms for k-anonymity. In Section 6.1, we describe an algorithm for (k, k)-anonymization, and in Section 6.2 we describe an algorithm for transforming a (k, k)-anonymized table to one that satisfies (in addition to (k, k)-anonymity) also the k-concealment property.

6.1 (k, k)-Anonymization

In this section we describe algorithms for (k, k)-anonymizing a given database D. First, we present in Section 6.1.1 algorithms for $(k, 1)$-anonymization. Then, in Section 6.1.2, we describe an algorithm for transforming a $(k, 1)$-anonymization into a (k, k)-anonymization.

6.1.1 Algorithms for $(k, 1)$-anonymization

Given a database $D = \{R_1, \dots, R_n\}$, Algorithm 1 finds an optimal $(k, 1)$-anonymization of D. It does so by finding for each record R_i, $1 \leq i \leq n$, the best subset of $k - 1$ additional records such that the generalization cost (see Definition 3) of those records, together with R_i, would be minimal. Once that subset is found, $\overline{R_i}$ is set to be the corresponding generalized record.

Proposition 8. *Algorithm 1 produces a table $g(D)$ that is an optimal $(k, 1)$-anonymization of D.*

The proof of proposition 8 is straightforward: the generalization $g(D)$ is a $(k, 1)$-anonymization of D because each record in it is a closure of k records in D. Its cost is optimal because when composing each generalized record $\overline{R_i}$, the exhaustive search is performed on the whole database D (and not only on a subset of not yet chosen records). This can be done because the iterations of the algorithm are independent of each other.

Algorithm 1 Optimal algorithm for $(k, 1)$-anonymization

```
Input: Table $D$, integer $k$.
Output: Table $g(D)$ that satisfies $(k, 1)$-anonymity.
for all $1 \leq i \leq n$ do
    For all $\binom{n-1}{k-1}$ selections of $k-1$ records, $R_{i_1}, \ldots, R_{i_{k-1}}$, out of $D \setminus \{R_i\}$, compute the generalization cost $d(\{R_i, R_{i_1}, \ldots, R_{i_{k-1}}\})$.
    Let $\{R_{i_1}, \ldots, R_{i_{k-1}}\}$ be a selection that resulted in a minimal generalization cost in the previous step.
    Define $\overline{R}_i$ to be the closure of $\{R_i, R_{i_1}, \ldots, R_{i_{k-1}}\}$.
end for
```

The runtime of Algorithm 1 is $O(n \cdot \binom{n-1}{k-1}) = O(n^k)$. Even though polynomial in the large parameter n, it is impractical because of the exponential dependence on k. Algorithm 2, which we describe below, constructs the generalized records by greedily selecting at each stage the next closest record. Its runtime is $O(kn^2)$ and it offers an approximation ratio guarantee of $k-1$ (see Appendix B).

Algorithm 2 $(k, 1)$-Anonymization by expansion

```
Input: Table $D$, integer $k$.
Output: Table $g(D)$ that satisfies $(k, 1)$-anonymity.
for all $1 \leq i \leq n$ do
    Set $S_i = \{R_i\}$
    while $|S_i| < k$ do
        Find the record $R_j \notin S_i$ that minimizes dist$(S_i, R_j) = d(S_i \cup \{R_j\}) - d(S_i)$.
        Set $S_i = S_i \cup \{R_j\}$.
    end while
    Define $\overline{R}_i$ to be the closure of $S_i$.
end for
```

6.1.2 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. Such a generalization may not satisfy $(1, k)$-anonymity since there could be records $R_i \in D$ that are consistent with less than k generalized records in $g(D)$. Algorithm 3 find such records in D and then it further generalizes the records of $g(D)$ until it becomes a $(1, k)$-anonymization of D. Specifically, if a given record R_i is consistent with only $\ell < k$ records in $g(D)$, the algorithm searches for additional $k - \ell$ generalized records in $g(D)$ that could be further generalized in order to become consistent with R_i with minimal cost. To simplify notations, 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 $\overline{R}_j$.

By applying this algorithm to a generalization that is already $(k, 1)$-anonymized, we end up with a generalized table $g(D)$ that respects (k, k)-anonymity.

The runtime of Algorithm 3 is $O(n^2)$: The outer loop consists of n steps; then for each record we need to check with how many of the n generalized records in $g(D)$ it is consistent, and if that number is less than k, we have to find the $k - \ell$ best ones ($O(kn)$ operations) and replace up to k generalized records. Hence, the overall runtime is $O(kn^2)$. Consequently,

Algorithm 3 $(1, k)$-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.
for all $1 \leq i \leq n$ do
    Let $\ell$ be the number of records $\overline{R}_j$ that are consistent with $R_i$.
    if $\ell < k$ then
        Scan all records $\overline{R}_j$ that are not consistent with $R_i$ and find the $k - \ell$ ones that
        minimize $c(R_i + \overline{R}_j) - c(\overline{R}_j)$.
        Replace each of those $k - \ell$ records, $\overline{R}_j$, with $R_i + \overline{R}_j$.
    end if
end for
```

so is the runtime of the coupling of that algorithm with the $(k, 1)$-anonymizer, Algorithm 2 (such a coupling is a (k, k)-anonymizer).

6.2 An algorithm for k-concealment

Next, we describe Algorithm 5 that transforms a (k, k)-anonymization $g(D)$ of a given database D into a k-concealed table. In order to understand the main idea behind the algorithm, we begin by characterizing the set of all matches in the graph (see Definition 6).

6.2.1 Finding all matches in a bipartite graph

Let $G = (U, V, E)$ be a bipartite graph where $U = \{u_1, \ldots, u_n\}$, $V = \{v_1, \ldots, v_n\}$ and $E \subseteq U \times V$. Assume also that G has at least one perfect matching; for the sake of convenience we assume that the perfect matching is $\{(u_1, v_1), \ldots, (u_n, v_n)\}$. In the spirit of Definition 6, an edge in E is called a *match* if it may be extended to a perfect matching in G. In Proposition 10 below we characterize the set of all matches in G.

Definition 9. A set of $\ell \geq 1$ edges in the graph G is called a *bicycle* (with respect to the assumed perfect matching) if there exist ℓ indices, $1 \leq i_1, \ldots, i_\ell \leq n$, such that the ℓ edges are

$$(u_{i_1}, v_{i_2}), (u_{i_2}, v_{i_3}), \ldots, (u_{i_{\ell-1}}, v_{i_\ell}), (u_{i_\ell}, v_{i_1}) . \tag{5}$$

It is important to note that a bicycle is not a cycle. For example, each of the edges (u_i, v_i), $1 \leq i \leq n$, is a bicycle of length $\ell = 1$, which is obviously not a cycle. Each bicycle of length $\ell > 1$ corresponds to a cycle of length 2ℓ; indeed, if we augment the bicycle in (5) with the ℓ "horizontal" edges (u_{i_j}, v_{i_j}), $1 \leq j \leq \ell$, we get a cycle. The converse, however, is not true. Consider for example the bipartite graph in Figure 4. The four non-horizontal edges are a cycle of length 4 that does not correspond to any bicycle; indeed, the only bicycles in that graph are the four bicycles of length 1 each (since the graph has only one perfect matching).

Proposition 10. *Let $G = (U, V, E)$ be a bipartite graph where $U = \{u_1, \ldots, u_n\}$, $V = \{v_1, \ldots, v_n\}$ and $E \subseteq U \times V$, and assume that $\{(u_1, v_1), \ldots, (u_n, v_n)\} \subset E$. Then an edge $e \in E$ is a match if and only if it is a part of some bicycle.*

Proof. Let us assume first that $e = (u_{i_1}, v_{i_2})$ is part of a bicycle, say

$$(u_{i_1}, v_{i_2}), (u_{i_2}, v_{i_3}), \ldots, (u_{i_{\ell-1}}, v_{i_\ell}), (u_{i_\ell}, v_{i_1}) .$$

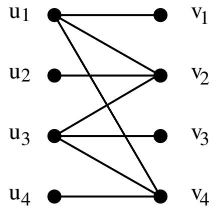

Figure 4: A bipartite graph with no nontrivial bicycles

If we augment this bicycle with the $n-\ell$ edges (u_i, v_i) for all $i \notin \{i_1, \ldots, i_\ell\}$ we get a perfect matching. The proof in the other direction is immediate since any perfect matching may be expressed as a disjoint union of bicycles. □

Let $G = (U, V, E)$ be the bipartite graph that corresponds to some anonymization $g(D)$ of D; namely, $U = \{R_1, \ldots, R_n\}$ consists of all original records and $V = \{\overline{R}_1, \ldots, \overline{R}_n\}$ consists of all generalized records. Since $\overline{R}_i$ is a generalization of R_i there is an edge that connects $u_i = R_i$ to $v_i = \overline{R}_i$ for all $1 \leq i \leq n$, and all of those edges are matches. We aim now to find all other edges of E that are matches. To that end we proceed as follows. First, we define the directed graph $H = (U, F)$ that is induced by the bipartite graph $G = (U, V, E)$. In the directed graph $H = (U, F)$ the set of nodes is $U = \{u_1, \ldots, u_n\}$ and $(u_i, u_j) \in F$ if and only if $i \neq j$ and $(u_i, v_j) \in E$. It is easy to see that, in view of Proposition 10, an edge $(u_i, v_j) \in E$ is a match in G if and only if $i = j$ or the edge $(u_i, u_j) \in F$ is part of a cycle in H. Hence, the problem of finding all matches in G reduces to the problem of finding all edges in the directed graph H that are part of a cycle. This may be achieved as follows: First, one has to find all strongly connected components of H; namely, all maximal strongly connected subgraphs of H (a directed graph is strongly connected if there is a path from each node in the graph to every other node). If each strongly connected component is contracted to a single node, the resulting graph is a directed acyclic graph. Consequently, a given edge in H is a part of cycle if and only if it connects two nodes in the same strongly connected component.

Algorithm 4 Finding all matches in a bipartite graph $G = (U, V, E)$.

```
Input: A bipartite graph $G = (U, V, E)$ where $U = \{u_1, \ldots, u_n\}$, $V = \{v_1, \ldots, v_n\}$, $E \subseteq U \times V$, and for all $1 \leq i \leq n$, $(u_i, v_i) \in E$.
Output: Marking all edges in $E$ by either YES or NO to indicate whether they are matches in $G$.
1: Construct the directed graph $H = (U, F)$ that corresponds to $G$.
2: Find all strongly connected components of $H$.
3: For all edges $(u_i, u_j) \in F$, if $u_i$ and $u_j$ belong to the same strongly connected component in $H$, mark the edge $(u_i, v_j) \in E$ as YES, otherwise mark it as NO.
4: Mark all edges $(u_i, v_i) \in E$, $1 \leq i \leq n$, as YES.
```

There are several efficient algorithms for finding the strongly connected components of a given directed graph. Tarjan's algorithm [40] and Cheriyan-Mehlhorn-Gabow algorithm [11] are both equally efficient with a linear runtime. We may apply one of those algorithms, and then identify all edges $(u_i, u_j) \in F$ that are part of a cycle in H: (u_i, u_j) is a part of a cycle if and only if u_i and u_j belong to the same strongly connected component. Finally, we use those findings to mark all matches in the original bipartite graph G. Algorithm 4 does all of the above. Its runtime is $O(|U| + |E|)$.

6.2.2 The algorithm

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 $|Q| \geq k$, but $|P|$ could be less than k. In order to achieve k-concealment, 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 generalize the record $\overline{R}_i$ to be consistent also with R_{j_h}. The discussion in Section 6.2.1 implies that this update of $\overline{R}_i$ "upgrades" $\overline{R}_{j_h}$ from a mere neighbor of R_i to a match of R_i (since now the edge $(R_i, \overline{R}_{j_h})$ becomes a part of a bicycle of length 2). This upgrade of the edge $(R_i, \overline{R}_{j_h})$ into being a match may have a similar effect on other edges as well. Hence, we recompute the set of matches and repeat the procedure until $|P|$ becomes at least k. Once this is accomplished, we move on to deal with the next node R_{i+1}.

Algorithm 5 From (k, k)-anonymity to k-concealment

```
Input: Table $D = \{R_1, \ldots, R_n\}$, and a 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: A further generalization of $g(D)$ that is $k$-concealed (as well as $(k, k)$-
    anonymized).
1:  Find all matches in the graph $V_{D,g(D)}$ (Algorithm 4).
2:  for all $1 \leq i \leq n$ do
3:      Set $Q = \{\overline{R}_{j_1}, \ldots, \overline{R}_{j_q}\}$ to be the set of $q \geq k$ neighbors of $R_i$.
4:      Extract $P$ – the subset of $Q$ consisting of all matches of $R_i$.
5:      If $|P| \geq k$, skip to next $i$. Otherwise, proceed with steps 6-10.
6:      For all $1 \leq h \leq q$ such that $\overline{R}_{j_h} \in Q \setminus P$, compute $d_h = c(R_{j_h} + \overline{R}_i) - c(\overline{R}_i)$.
7:      Select the index $1 \leq h \leq q$ where $\overline{R}_{j_h} \in Q \setminus P$, for which $d_h$ is minimal.
8:      Set $\overline{R}_i = R_{j_h} + \overline{R}_i$.
9:      Recompute the set of all matches in the graph $V_{D,g(D)}$.
10:     Return to Step 3.
11: end for
```

Algorithm 5 invokes Algorithm 4 once at the beginning, before it starts the main loop. Then, after each addition of a new edge (Step 8), it needs to recompute the matches in the graph (Step 9), since the addition of a new edge might upgrade more than one edge into being a match. In order to do that efficiently, we keep the partition of the directed graph $H = (U, F)$ to strongly connected components by keeping $H' = (U', F')$ — the directed graph in which $U' = \{C_1, \ldots, C_m\}$ is the set of strongly connected components in H, and F' has an edge from C_a to C_b if F has an edge from a node in C_a to a node in C_b. Now, the operation in Step 8 in Algorithm 5 is equivalent to adding the edge (u_{j_h}, u_i) to H. Since those two nodes belong to two distinct strongly connected components in H, say $u_{j_h} \in C_1$ and $u_i \in C_2$, the operation in Step 8 results also in adding a new edge (C_1, C_2) to H'. Hence, what we need to do is to apply Tarjan's algorithm on the smaller graph H' and find the strongly connected components in it at this stage. The new strongly connected components in H' tell us how to update the separation of H into strongly connected components and which edges in H upgrade into being matches.

In order to avoid unnecessary formalism, we prefer to illustrate the above ideas by an example. Assume that H has 5 strongly connected components and we added in Step 8 an edge (u_{j_h}, u_i), where $u_{j_h} \in C_1$ and $u_i \in C_2$. Then we add the edge (C_1, C_2) to H' and then run Tarjan's algorithm on H'. Assume that we find out that the strongly connected components in H', in wake of this edge addition, are $\{C_1, C_2, C_3\}$, $\{C_4\}$, $\{C_5\}$. Then we conclude that H now has only 3 strongly connected components — $C_1 \cup C_2 \cup C_3$, C_4, C_5. Moreover, all edges in H that connect a node from C_i to a node in C_j, where $1 \leq i \neq j \leq 3$, will now be matches.

We note that the least that could happen in wake of the operation in Step 8 is the unification of C_1 and C_2 (where $u_{j_h} \in C_1$ and $u_i \in C_2$). Indeed, since H already has the edge in the opposite direction, (u_i, u_{j_h}), (namely, H' has the edge (C_2, C_1)), then the new edge (C_1, C_2) makes $\{C_1, C_2\}$ strongly connected in H'. But, since it is possible that the addition of that single edge will create a larger strongly connected component we have to apply Tarjan's algorithm on H'.

The runtime of Algorithm 5 is analyzed as follows. Finding all matches in Step 1 by invoking Algorithm 4 takes $O(kn)$ time. Then, we start a loop over all n nodes, and for each node we perform at most k upgrades of an edge to a match. By keeping an array that holds for each node the connected component to which it belongs, we can decide in $O(1)$ whether an edge in the graph is a match. Steps 3–8 require $O(k)$ time and Step 9 requires $O(nk)$ time. The overall runtime of Algorithm 5 is therefore $O(kn^2)$.

7 Randomizing the algorithms

A basic assumption in cryptography is Kerckhoffs' principle [25]. It states that the security of a cryptographic algorithm must not rely on the assumed secrecy of the algorithm, but only on the random selections (namely, the key) that the algorithm uses. Hence, when designing a cryptographic algorithm it must be assumed that the algorithm is known to the adversary.

As we assumed that the adversary knows D as well as $g(D)$, and that he knows also the anonymization algorithms, he may construct the graph $V_{D,g(D)}$. Therefore, he knows for each original record R_i what is the value of its generalization $\overline{R}_i$. Hence, given an original record R_i, the adversary would know that its generalization is one of the generalized records in $g(D)$ that equal $\overline{R}_i$. The number of those suspect records may be, in the worst case, one.

Two previous studies have also identified this risk in the context of anonymizing tables [46, 49]. Our approach here is closer to the one in [46]. The problem that they identified is when the data owner attempts to ℓ-diversify a table with minimal loss of information. They showed that since the partition of the table records into equivalence classes of records is guided by the need to respect ℓ-diversity (a condition which depends only on the sensitive attribute) and also by the goal to avoid unnecessary generalizations, it is possible sometimes to infer the sensitive values of some of the records. They identified that even though k-anonymity on its own does not provide a sufficient level of privacy, since it does not consider the sensitive values when making decisions on how to partition the table records, it is exactly that feature that makes it a useful component in thwarting such minimality attacks. Their proposed algorithm MASK (Minimality Attack Safe K-anonymity) has two phases. First, it k-anonymizes the table. Then, they execute further generalizations in order to make sure that all equivalence classes satisfy ℓ-diversity. The second stage uses random decisions as part of the blinding effect.

Our algorithm works also in two phases: A first phase, that achieves k-concealment and does not consider the sensitive value, followed by the second phase that achieves ℓ-diversity or p-sensitivity in addition to k-concealment (the second phase is described in the next section). We introduce randomization in both phases. Herein we describe the randomization of the k-concealment phase.

k-Concealment is achieved in three steps. First, we find a $(k,1)$-anonymization of D, using Algorithm 2. Then we transform the $(k,1)$-anonymization into a (k,k)-anonymization (Algorithm 3) and then to a k-concealment (Algorithm 5). We introduce randomization in all three steps.

In the first step, we consider a randomized version (Algorithm 6) of Algorithm 2. That version produces, for each $1 \leq i \leq n$, two different generalized records, $\tilde{R}_0$ and $\tilde{R}_1$, each of which is consistent with R_i and at least $k-1$ other records in D. Then, we select at random one of those two generalized records and define the ith record in $g(D)$ as that generalized record. The first generalized record, $\tilde{R}_0$, is the one that Algorithm 2 would produce (Step 4). In order to find a second generalized record, we look at the q closest neighbors of R_i (where $q > k-1$) and then randomly select $k-1$ out of them in order to define the second generalized record $\tilde{R}_1$, see Step 5. If $\tilde{R}_0 \neq \tilde{R}_1$, we define $\overline{R}_i$ to be one of them with equal probabilities (Step 7). If $\tilde{R}_0 = \tilde{R}_1$, we repeat our random selection until a different generalized record is obtained. (We may limit the number of trials, and then, if all trials failed, we can select $\tilde{R}_1$ to be any random generalization of $\tilde{R}_0$.) Since an independent selection is made for each record, the overall number of generalizations $g(D)$ that could be the output of this phase is at least 2^n. (In fact, since $\tilde{R}_1$ is random in itself, as there are $\binom{q}{k-1}$ ways to define it, the number of possibilities that needs to be checked by the adversary is even larger.) Hence, the adversary would need to simulate the next two steps in the k-concealment process (Algorithms 3 and 5) on at least 2^n graphs $g(D)$ that could be the output of the first step.

Algorithm 6 Randomized $(k,1)$-anonymization

```
Input: Table $D$, integer $k$, and integer $q > k-1$.
Output: Table $g(D)$ that satisfies $(k,1)$-anonymity.
For all $1 \leq i < j \leq n$ compute $d_{i,j} = d_{j,i} = d(\{R_i, R_j\})$.
for all $1 \leq i \leq n$ do
    Find $q$ indices $\mathcal{J}_q := \{j_1, \ldots, j_q\}$ that minimize $d_{i,j}$ in $\{1, \ldots, n\} \setminus \{i\}$. (We order those indices so that $d_{i,j_1} \leq \cdots \leq d_{i,j_q}$.)
    Define $\tilde{R}_0$ to be the generalization of $R_i$ as computed by Algorithm 2.
    Randomly select $k-1$ indices out of $\mathcal{J}_q$, and then define $\tilde{R}_1$ to be the closure of the subset of records consisting of the corresponding $k-1$ records and $R_i$.
    Repeat Step 5 until $\tilde{R}_1 \neq \tilde{R}_0$.
    Select a random bit $b \in \{0,1\}$ and set $\overline{R}_i = \tilde{R}_b$.
end for
```

Further randomization may be introduced in the next two steps of achieving k-concealment. To do that, we select a random ordering of the records in the main loop of Algorithm 3 and an independent random ordering of the records in the main loop of Algorithm 5. In each of those algorithms we scan all records $R_1, \ldots, R_n$ in the table and check whether they satisfy some anonymity condition ($(1,k)$-anonymity in Algorithm 3 and k-concealment in Algorithm 5), and if they do not, we mend the problem by performing appropriate generalizations. The order in which the records are visited has an effect on the final output since generalizations that are done in one step of the loop may have an effect on the need to perform generalizations in later steps. By performing the main loop in each of those algorithms in a random and independent order, we introduce a significant factor of randomness ($(n!)^2$), which multiplies the previous randomness factor of 2^n.

8 Enhancing k-concealment by diversity measures

As discussed in Section 1.2, k-anonymized tables must respect also an additional privacy measure such as ℓ-diversity or p-sensitivity. Here, we describe how our algorithms may be enhanced towards that end.

Let $D = \{R_1, \ldots, R_n\}$ denote the original table and $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ be the generalized table that respects k-concealment. The latter table is released together with the private data; namely, each generalized record, $\overline{R}_i$, is coupled with the corresponding private attribute s_i, $1 \le i \le n$. Assume that R_i has t matches in $g(D)$; let us denote them $\overline{R}_{i_j}$, where $1 \le j \le t$. Then the adversary may deduce that the private value of R_i is one of the private values in the multi-set $S_i = \{s_{i_1}, \ldots, s_{i_t}\}$. The goal is to have all multi-sets S_i, $1 \le i \le n$, satisfy some diversity condition. The two conditions that we shall discuss here are:

- p-sensitivity: S_i must contain at least p different values.
- ℓ-diversity: The most frequent value in S_i must have a frequency of no more than t/ℓ.

This goal may be achieved by further generalization. Namely, we look for a generalization $g'(D) = \{\overline{R}'_1, \ldots, \overline{R}'_n\}$ of D, where $\overline{R}'_i$ either equals $\overline{R}_i$ or generalizes it, $1 \le i \le n$, that respects p-sensitivity or ℓ-diversity in the sense described above. Clearly, as $g'(D)$ generalizes $g(D)$ and $g(D)$ is a k-concealment of D, so is $g'(D)$. We look for such $g'(D)$ with as low as possible further generalization cost.

The idea is to check those multi-sets to see whether some of them fail to satisfy the required diversity condition. Assume that S_i is not sufficiently diverse. Then we select, in a greedy manner, a minimal number of generalized records in $g(D)$ that are not matches of R_i, and then turn them into matches of R_i, by further generalizations. As a result, the corresponding sensitive values of those new matches will be added to S_i. By properly selecting the new matches, those added sensitive values will turn S_i into a sufficiently diverse multi-set.

Recall that in order to turn a generalized record $\overline{R}_j$ into a match of R_i we can create a bi-cycle of length 2 between them. Namely, we can replace $\overline{R}_i$ with $R_j + \overline{R}_i$ (i.e., further generalize $\overline{R}_i$ until it becomes consistent also with R_j) and replace $\overline{R}_j$ with $R_i + \overline{R}_j$. In order to extend the set of matches of R_i greedily, we can always select a generalized record $\overline{R}_j$ that would get us closer to our goal (namely, the corresponding sensitive value s_j will enrich S_i towards the required diversity) and for which the required generalization cost is minimal.

It remains only to describe how to select a minimal subset of non-matches of R_i that should be matched with R_i.

8.1 Achieving p-sensitivity

If S_i contains only $p' < p$ different values, we greedily pick non-match generalized records $\overline{R}_j$ that have different private values that do not appear in S_i, and turn them into matches of R_i, until S_i becomes p-sensitive. We add those new matches one at a time. After adding one edge in order to create a new match for R_i, we recompute the set of matches in the graph (an operation which is very simple, as explained in Section 6.2.2 regarding Step 9 in Algorithm 5); this is done since it is possible that the addition of just one edge will add more than one match to R_i. Algorithm 7 implements that procedure. (We include in the input to that algorithm the complete set of all matches in the corresponding bipartite graph since Algorithm 5 computes and updates that set during its execution, and therefore it can transfer it as an input to Algorithm 7.)

Algorithm 7 Enhancing k-concealment by p-sensitivity

```
Input: • Tables $D, g(D)$, where $g(D)$ is a $k$-concealment of $D$;
       • The set of all matches in the bipartite graph $V_{D,g(D)}$;
       • An integer $p$.
Output: Table $g(D)$ that satisfies both $k$-concealment and $p$-sensitivity.
 1: for all $1 \leq i \leq n$ do
 2:   Let $M_i := \{\overline{R}_{i_1}, \ldots, \overline{R}_{i_t}\}$ be the set of matches of $R_i$ and $S_i := \{s_{i_1}, \ldots, s_{i_t}\}$ be their
      sensitive values.
 3:   Compute the number $p'$ of different values in $S_i$.
 4:   while $p' < p$ do
 5:     Find a record $\overline{R}_j \notin M_i$ for which $s_j \notin S_i$, that minimizes $[c(R_i + \overline{R}_j) - c(\overline{R}_j)] +$
        $[c(R_j + \overline{R}_i) - c(\overline{R}_i)]$.
 6:     $\overline{R}_j = R_i + \overline{R}_j$ .
 7:     $\overline{R}_i = R_j + \overline{R}_i$ .
 8:     Update the set of matches in the graph.
 9:     Recompute $M_i$, $S_i$ and $p'$.
10:   end while
11: end for
```

8.2 Achieving ℓ-diversity

Algorithm 7 may be modified in order to achieve ℓ-diversity instead of p-sensitivity; we concentrate on the interpretation of diversity as the inverse of the maximal frequency in S_i. To that end, we shall compute in Step 3 of the algorithm the diversity ℓ' of S_i and compare it to the required minimal diversity ℓ. While $\ell' < \ell$ we shall select a generalized record $\overline{R}_j$ for which s_j is different from the most frequent value in S_i and whose turning into a match of R_i entails the minimal generalization cost. The loop will continue until S_i becomes ℓ-diverse.

9 Experimental results

In this section we discuss the experiments that we performed in order to evaluate the new anonymity model of k-concealment and our proposed algorithms. We tested the algorithms on both artificial and real data.

Artificial data. The artificial database consisted of $n = 5000$ records over a set of six attributes $A_1, \ldots, A_6$. Each of those six attributes consisted of a finite number of 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 example, attribute A_1 has two possible values, the first of them with probability 0.7 and the second with probability 0.3. Attribute A_4, on the other hand, contains 6 values with probability 0.07, 10 values with probability 0.04, and 9 values with probability 0.02.

For each of the above attributes, $A = \{a_1, \ldots, a_m\}$, the collection of permissible generalized subsets, $\overline{A}$, is described below. As all of those collections include all singleton subsets, $\{a_i\}$, $1 \le i \le m$, 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 $\{a_1, a_2\}$)
$\overline{A}_2 : \{a_1, a_2\}, \{a_3, a_4\}$
$\overline{A}_3 : \{a_1, a_2\}, \{a_3, a_4\}$
$\overline{A}_4 : \{a_1, \ldots, a_6\}, \{a_7, \ldots, a_{12}\}, \{a_{13}, \ldots, a_{18}\},$
$\{a_{19}, \ldots, a_{25}\}, \{a_1, \ldots, a_{12}\}, \{a_{13}, \ldots, a_{25}\}$
$\overline{A}_5 : \{a_1, a_2\}, \{a_3, a_4\}, \{a_6, a_7\}, \{a_8, a_9\},$
$\{a_1, a_2, a_3, a_4, a_5\}, \{a_6, a_7, a_8, a_9, a_{10}\}$
$\overline{A}_6 : \{a_1, a_2\}, \{a_4, a_5\}, \{a_3, a_4, a_5\}$

Note, for example, that $\overline{A}_6$ defines an unbalanced tree of height 3. The distance of a_4 and a_5 from the root $\{a_1, a_2, a_3, a_4, a_5\}$ is 3, while the distance of a_1, a_2 and a_3 from the root is 2.

Real-life data. We used three real-life datasets, ADULT, CONTRACEPTIVE METHOD CHOICE (or CMC), and NURSERY from the UCI Machine Learning [15].

ADULT: This dataset was extracted from the US Census Bureau Data Extraction System. It contains demographic information of a small sample of the US population ($n = 45,222$) with 14 public attributes such as `age`, `education-level`, `marital-status`, `occupation`, and `native-country`. The private information is an indication whether that individual earns more or less than 50 thousand dollars annually. (That dataset is commonly used in studies of anonymity, e.g. [6, 28, 35].) The collection of permissible generalized subsets in each of the attributes was selected by grouping together values that are semantically close.

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 1500 records and 9 public attributes.

NURSERY: This dataset was derived from a hierarchical decision model that was originally developed to rank applications for nursery schools. The NURSERY dataset contains 12960 records after deleting those with missing values. It has 8 public attributes.

The algorithms were implemented in Java and ran on Pentium® 4, CPU 3.00 GHz, 960MB of RAM.

In our first set of experiments we examined the information losses as achieved by k-anonymity algorithms and our k-concealment algorithm. The two k-anonymity algorithms that we used were the agglomerative algorithm [19] and the forest algorithm of Aggarwal et al. [2]. The algorithms were tested for each combination of the entropy measure (EM), [20], and the Loss Metric measure (LM) (see Eqs. (1)+(2)) with the four datasets as described above, for k ranging from $k = 10$ to $k = 100$. The results are given in Figures 5—8.

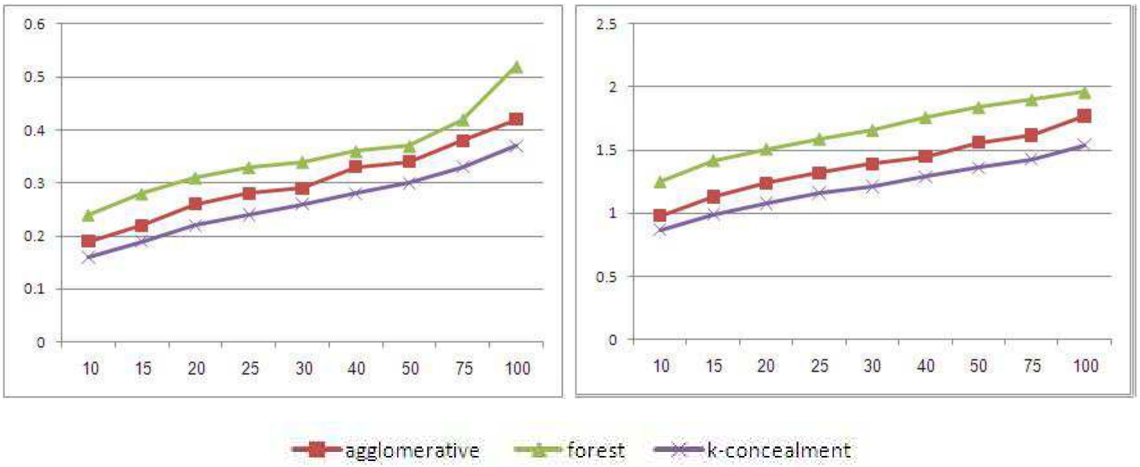

Figure 5: Information losses in the artificial dataset — LM (left) and EM (right)

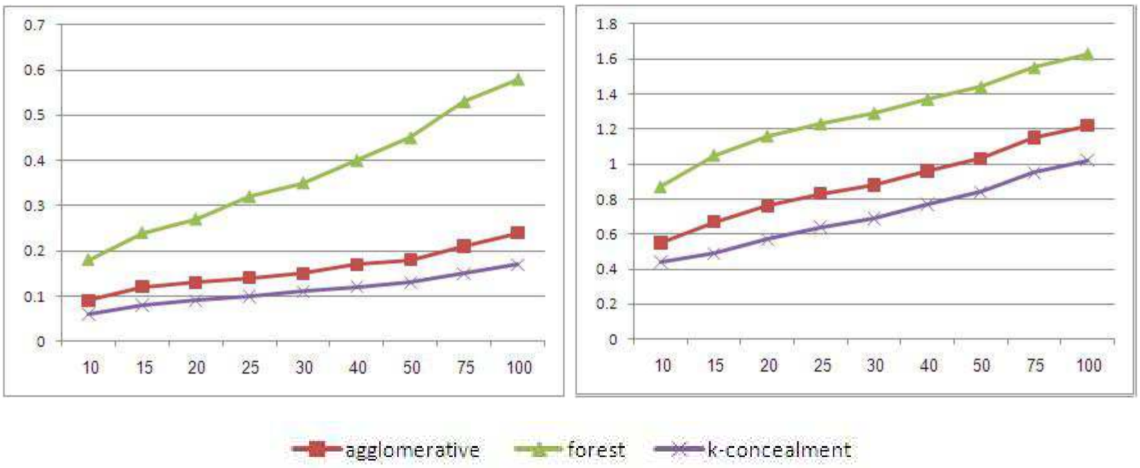

Figure 6: Information losses in ADULT dataset — LM (left) and EM (right)

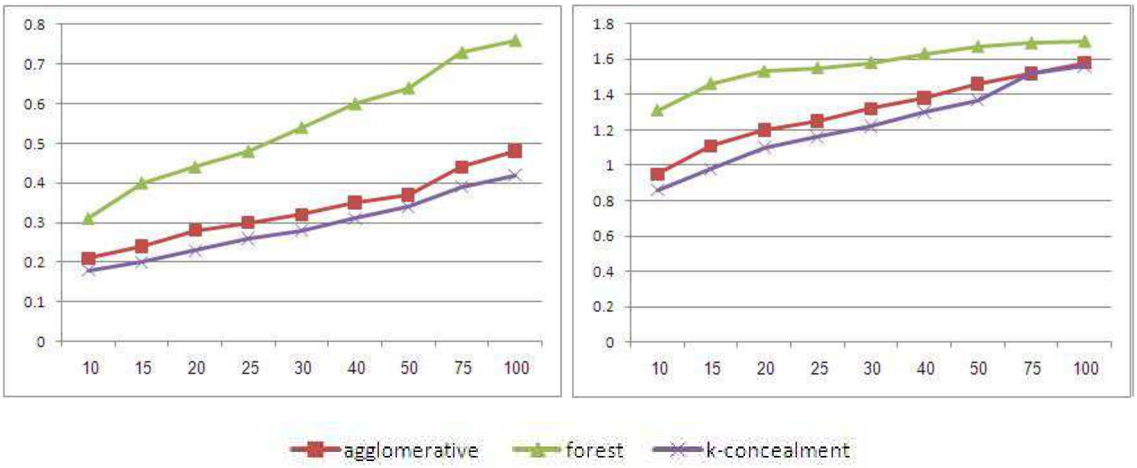

Figure 7: Information losses in CMC dataset — LM (left) and EM (right)

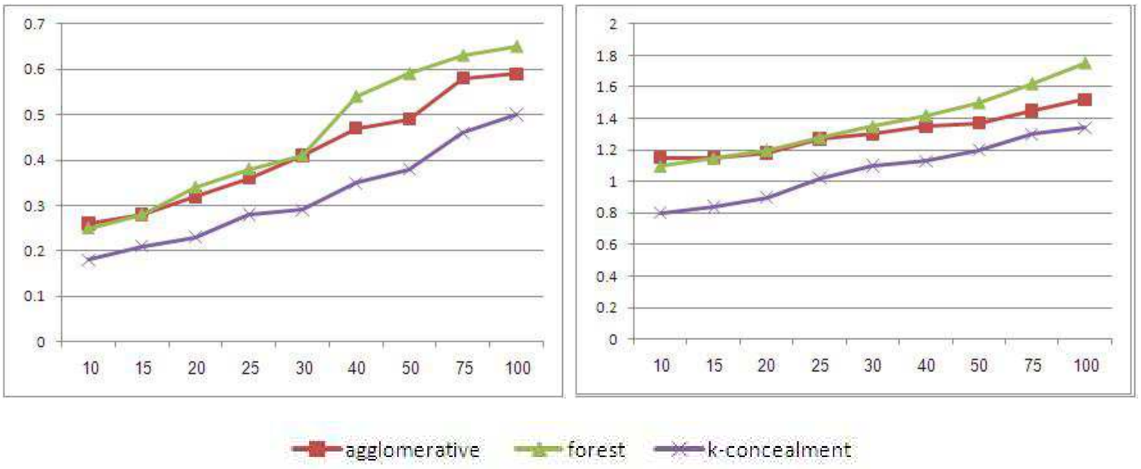

Figure 8: Information losses in NURSERY dataset — LM (left) and EM (right)

As expected, k-concealment offers smaller information losses than k-anonymity, in all datasets and for the two information loss measures that were tested.

Next, we tested the randomized version of our k-concealment algorithm (where Algorithm 6 is used in the first step of producing a $(k, 1)$-anonymization) in order to assess the degradation in the utility of the anonymized tables that are produced by that algorithm. We used the parameter $q = 2(k - 1)$ in Algorithm 6. Figure 9 shows the information losses in the agglomerative algorithm, the basic k-concealment algorithm and the randomized one, on the ADULT dataset. The randomized algorithm, which makes on purpose randomal non-optimal decisions, yields information losses that are larger than those of the basic k-concealment algorithm, as expected, but are still better than those of the agglomerative algorithm.

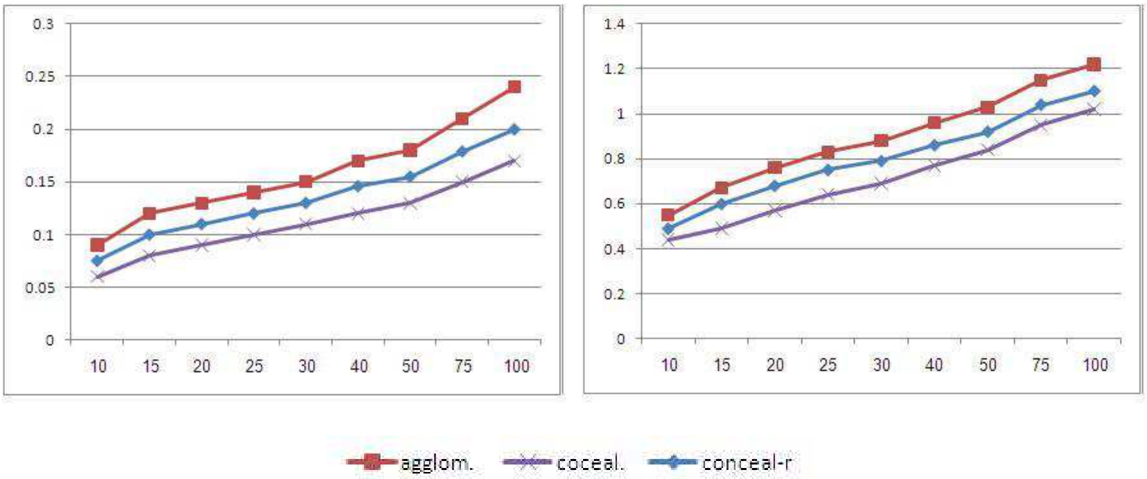

Figure 9: Testing the effect of randomization on the information losses — ADULT dataset; LM (left) and EM (right)

In our last set of experiments we tested the version of the k-concealment algorithm that is designed to achieve ℓ-diversity, as described in Section 8. We used it in the ADULT dataset, with $\ell = 1.1$, and in the NURSERY dataset, with $\ell = 2$. (The maximal possible diversities for the ADULT and NURSERY datasets are 1.333 and 3, respectively.) For the sake of comparison, we tested also a modification of the agglomerative algorithm that respects ℓ-diversity (see details in Appendix C). The LM information losses in both experiments are shown in Figure 10. As can be seen, the advantage of k-concealment in terms of the information loss in the resulting anonymization is preserved also when considering ℓ-diversity as an additional constraint.

10 Conclusions

In this paper we proposed the model of k-concealment as an alternative to k-anonymity. We showed that k-concealment offers essentially the same level of security as k-anonymity: While k-anonymity ensures that every record in the released table is identical to at least $k-1$

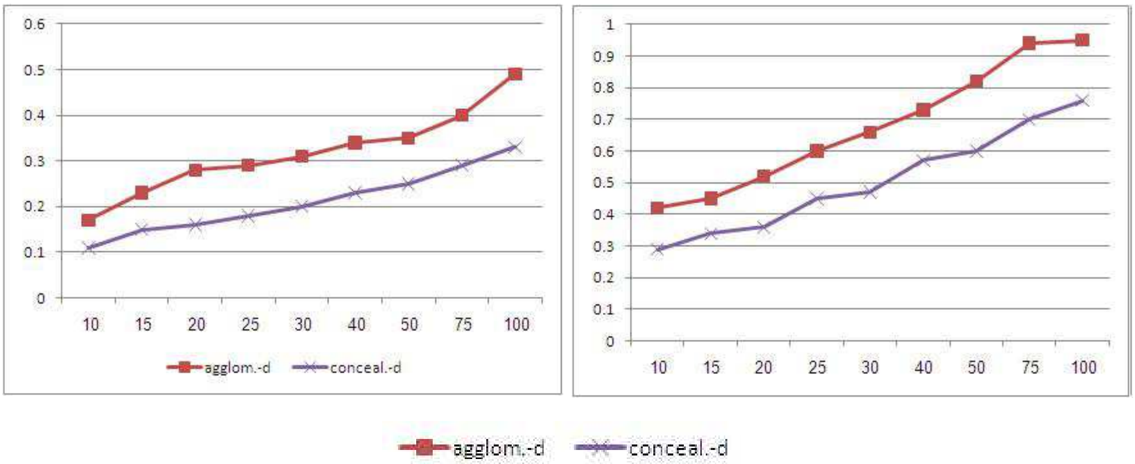

Figure 10: Information losses in the diversified algorithms — ADULT (left) and NURSERY (right)

others, k-concealment guarantees that every record is computationally indistinguishable from at least $k-1$ others. The advantage that is offered by k-concealment is that it allows reducing the information loss that is caused by generalizing the database entries. Hence, k-concealed tables offer more utility than k-anonymized tables, as demonstrated by our experiments, while providing a comparable level of security. Since k-anonymity on its own is not sufficiently secure and should be enhanced by additional measures of security that depend on the private attribute, such as ℓ-diversity, so does k-concealment. We described algorithms for achieving k-concealment and then described how to turn the k-concealed tables into ones that respect also p-sensitivity or ℓ-diversity.

Acknowledgements. The authors thank Erez Shmueli for suggesting an improvement to an earlier version of Algorithm 5.

References

[1] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu. Achieving anonymity via clustering. In *ACM-SIGMOD Symposium on Principles of Database Systems (PODS)*, pages 153–162, 2006.

[2] 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.

[3] G. Aggarwal, N. Mishra, and B. Pinkas. Secure computation of the kth-ranked element. In *EUROCRYPT*, pages 40–55, 2004.

[4] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *ACM-SIGMOD International Conference on Management of Data (SIGMOD)*, pages 439–450, 2000.

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

[6] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *International Conference on Data Engineering (ICDE)*, pages 217–228, 2005.

[7] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *ACM-SIGMOD Symposium on Principles of Database Systems (PODS)*, pages 128–138, 2005.

[8] B. Bollobás. *Graph Theory: An Introductory Course*. Springer-Verlag, 1979.

[9] L. Burnett, K. Barlow-Stewart, A. Proos, and H. Aizenberg. The" GeneTrustee": A universal identification system that ensures privacy and confidentiality for human genetic databases. *Journal of Law and Medicine*, 10(4):506, 2003.

[10] J. Cai, A. Pavan, and D. Sivakumar. On the hardness of permanent. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 90–99, 1999.

[11] J. Cheriyan and K. Mehlhorn. Algorithms for dense graphs and networks on the random access computer. *Algorithmica*, 15:521–549, 1996.

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

[13] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *ACM-SIGMOD Symposium on Principles of Database Systems (PODS)*, pages 211–222, 2003.

[14] U. Feige and C. Lund. On the hardness of computing the permanent of random matrices. In *ACM Symposium on Theory of Computing (STOC)*, pages 643–654, 1992.

[15] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[16] B. Fung, K. Wang, R. Chen, and P. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (CSUR)*, 42(4):1–53, 2010.

[17] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *ACM Symposium on Theory of Computing (STOC)*, pages 32–42, 1991.

[18] P. Gemmell and M. Sudan. Highly resilient correctors for polynomials. *Inf. Process. Lett.*, 43:169–174, 1992.

[19] A. Gionis, A. Mazza, and T. Tassa. k-Anonymization revisited. In *International Conference on Data Engineering (ICDE)*, pages 744–753, 2008.

[20] A. Gionis and T. Tassa. k-Anonymization with minimal loss of information. *IEEE Transactions on Knowledge and Data Engineering*, 21:206–219, 2009.

[21] J. Goldberger and T. Tassa. Efficient anonymizations with enhanced utility. *Transactions on Data Privacy*, 3:149–175, 2010.

[22] V. Iyengar. Transforming data to satisfy privacy constraints. In *ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 279–288, 2002.

[23] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51:671–697, 2004.

[24] B. Kenig and T. Tassa. A practical approximation algorithm for optimal k-anonymity. *Data Mining and Knowledge Discovery*, 2012.

[25] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, 9:5–83,161–191, 1883.

[26] D. Kifer and J. Gehrke. Injecting utility into anonymized datasets. In *ACM-SIGMOD International Conference on Management of Data (SIGMOD)*, pages 217–228, 2006.

[27] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Auditing boolean attributes. *Journal of Computer and System Sciences*, 6:244–253, 2003.

[28] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Incognito: efficient full-domain k-anonymity. In *ACM-SIGMOD International Conference on Management of Data (SIGMOD)*, pages 49–60, 2005.

[29] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *International Conference on Data Engineering (ICDE)*, page 25, 2006.

[30] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-

diversity. In *ICDE*, pages 106–115, 2007.

[31] R. Lipton. New directions in testing. *Distributed Computing and Cryptography, Vol. 2 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 191–202, 1991.

[32] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-Diversity: Privacy beyond k-anonymity. In *International Conference on Data Engineering (ICDE)*, page 24, 2006.

[33] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *ACM-SIGMOD Symposium on Principles of Database Systems (PODS)*, pages 223–228, 2004.

[34] M. E. Nergiz and C. Clifton. Thoughts on k-anonymization. *Data Knowl. Eng.*, 63(3):622–645, 2007.

[35] H. Park and K. Shim. Approximate algorithms for k-anonymity. In *ACM-SIGMOD Conference*, pages 67–78, 2007.

[36] H. Ryser. *Combinatorial Mathematics*. Carus Mathematical Monograph No. 14, Math. Assoc. of America, 1963.

[37] P. Samarati. Protecting respondent's privacy in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13:1010–1027, 2001.

[38] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In *ACM-SIGMOD Symposium on Principles of Database Systems (PODS)*, page 188, 1998.

[39] L. Sweeney. k-Anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[40] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[41] S. Toda. On the computational power of PP and $\oplus$P. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 514–519, 1989.

[42] T. Truta, A. Campan, and P. Meyer. Generating microdata with p-sensitive k-anonymity property. In *Secure Data Management (SDM)*, pages 124–141, 2007.

[43] J. Vaidya, Y. M. Zhu, and C. Clifton. *Privacy preserving data mining*. Springer-Verlag, 2006.

[44] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[45] S. Vinterbo. Privacy: a machine learning view. *IEEE Transactions on Knowledge and Data Engineering*, 16:939–948, 2004.

[46] R. Wong, A. Fu, K. Wang, and J. Pei. Minimality attack in privacy preserving data publishing. In *International Conference on Very Large Data Bases (VLDB)*, pages 543–554, 2007.

[47] R. Wong, J. Li, A. Fu, and K. Wang. (α, k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 754–759, 2006.

[48] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In *International Conference on Very Large Data Bases (VLDB)*, pages 139–150, 2006.

[49] L. Zhang, S. Jajodia, and A. Brodsky. Information disclosure under realistic assumptions: privacy versus optimality. In *ACM Conference on Computer and Communications Security*, pages 573–583, 2007.

Appendix

A Proving the interrelations between the classes of k-type anonymizations

Proof of Proposition 5. As all the inclusions in (3) are straightforward, it remains only to exemplify the inequalities in (3) and (4). Consider the following table D (having one attribute and $k+1$ records) alongside with four generalizations of it, taken from $\mathcal{A}_D^k$, $\mathcal{A}_D^{(1,k)}$, $\mathcal{A}_D^{(k,1)}$, and $\mathcal{A}_D^{(k,k)}$, respectively. (Here, $[k+1]$ stands for the set $\{1, 2, \ldots, k, k+1\}$.)

D	$g_k(D)$	$g_{(1,k)}(D)$	$g_{(k,1)}(D)$	$g_{(k,k)}(D)$
1	$[k+1]$	$\{1\}$	$[k+1] \setminus \{k+1\}$	$[k+1] \setminus \{1\}$
2	$[k+1]$	$[k+1]$	$[k+1] \setminus \{1\}$	$[k+1] \setminus \{2\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	$[k+1]$	$[k+1]$	$[k+1] \setminus \{1\}$	$[k+1] \setminus \{k\}$
$k+1$	$[k+1]$	$[k+1] \setminus \{1\}$	$[k+1] \setminus \{1\}$	$[k+1] \setminus \{k+1\}$

- The first generalization is obviously in $\mathcal{A}_D^k$ since all of the $k+1$ generalized records are equal.

- The second generalization is in $\mathcal{A}_D^{(1,k)}$; indeed, the first record in D is consistent with the first k generalized records, while each of the remaining records in D is consistent with the last k generalized records. However, that generalization is not in $\mathcal{A}_D^{(k,1)}$ since the first generalized record is consistent only with the first record in D.

- The third generalization is in $\mathcal{A}_D^{(k,1)}$ since each generalized record is consistent with exactly k records in D. However, that generalization is not in $\mathcal{A}_D^{(1,k)}$ since the first record in D is consistent only with the first generalized record.

- The last generalization is in $\mathcal{A}_D^{(k,k)}$, as can be easily seen. However, it is not in $\mathcal{A}_D^k$.

□

Proof of Proposition 7. As it is clear that $\mathcal{C}_D^k$ is a subset of $\mathcal{A}_D^{(1,k)}$ and a superset of $\mathcal{A}_D^k$, it remains only to prove that all six regions in Figure 1 are nonempty. One of those regions is $\mathcal{A}_D^k$, which is clearly nonempty, and another is $\mathcal{A}_D^{(k,1)} \setminus \mathcal{A}_D^{(1,k)}$ that was shown to be nonempty in Proposition 5 (see (4) there). The remaining four regions are:

- $\Omega_1 = \mathcal{A}_D^{(1,k)} \setminus \mathcal{A}_D^{(k,1)} \setminus \mathcal{C}_D^k$;

- $\Omega_2 = \mathcal{A}_D^{(k,k)} \cap \mathcal{C}_D^k \setminus \mathcal{A}_D^k$;

- $\Omega_3 = \mathcal{C}_D^k \setminus \mathcal{A}_D^{(k,1)}$;

- $\Omega_4 = \mathcal{A}_D^{(k,k)} \setminus \mathcal{C}_D^k$.

As in the proof of Proposition 5, we give specific anonymizations in each of those regions.

Consider the generalization $g_{(1,k)}(D)$ from the proof of Proposition 5. As shown there, it is in $\mathcal{A}_D^{(1,k)} \setminus \mathcal{A}_D^{(k,1)}$. We claim that it is in fact in Ω_1. To prove this claim, we need to show that it is not in $\mathcal{C}_D^k$. Indeed, the first record in D, denoted R_1, has only one match, which is the first generalized record in $g_{(1,k)}(D)$. Since the latter generalized record is consistent only with R_1, then the edge that connects those two records in the corresponding bipartite graph must be present in any perfect matching. That implies that all the other generalized records with which R_1 is consistent are not matches. Hence, R_1 has only one match and, consequently, $g_{(1,k)}(D) \notin \mathcal{C}_D^k$.

Consider next the generalization $g_{(k,k)}(D)$ from the proof of Proposition 5. As shown there, it is in $\mathcal{A}_D^{(k,k)} \setminus \mathcal{A}_D^k$. In order to prove that it is in Ω_2, we need to show that it is in $\mathcal{C}_D^k$. To that end, we demonstrate k perfect matchings in the corresponding bipartite graph, such that any given record in D is matched by those perfect matchings to k different records in $g_{(k,k)}(D)$. Denoting the records in D by R_i and those in $g_{(k,k)}(D)$ by $\overline{R}_i$, $0 \le i \le k$, the matching

$$(R_i, \overline{R}_{(i+\ell) \bmod (k+1)}) , \quad 0 \le i \le k$$

is a perfect matching for every value of $1 \le \ell \le k$.

Next, we exemplify the non-emptiness of Ω_3. Consider the following table D together with a corresponding generalization, $g(D)$.

D	$g(D)$
$R_1 = 1$	$\overline{R}_1 = \{1, 2, \ldots, k-1\}$
$R_2 = 2$	$\overline{R}_2 = \{1, 2, \ldots, k, k+1\}$
$\vdots$	$\vdots$
$R_k = k$	$\overline{R}_k = \{1, 2, \ldots, k, k+1\}$
$R_{k+1} = k+1$	$\overline{R}_{k+1} = \{1, 2, \ldots, k, k+1\}$

$g(D)$ is not in $\mathcal{A}_D^{(k,1)}$ since $\overline{R}_1$ is consistent with $k-1$ records. On the other hand, $g(D) \in \mathcal{C}_D^k$ as we proceed to show:

- R_1 has $k+1$ matches:
  - $\overline{R}_1$ is a match, as can be seen through the natural perfect matching, $\{(R_i, \overline{R}_i), 1 \le i \le k+1\}$.
  - $\overline{R}_j$, $2 \le j \le k-1$, is a match; the perfect matching is $\{(R_1, \overline{R}_j), (R_j, \overline{R}_1)\} \cup \{(R_i, \overline{R}_i)\}_{i \ne 1,j}$.
  - $\overline{R}_j$, $k \le j \le k+1$, is a match; the perfect matching is $\{(R_1, \overline{R}_j), (R_j, \overline{R}_2), (R_2, \overline{R}_1)\} \cup \{(R_i, \overline{R}_i)\}_{i \ne 1,2,j}$.
- R_j, $2 \le j \le k-1$, has $k+1$ matches:
  - $\overline{R}_1$ is a match; the perfect matching is $\{(R_1, \overline{R}_j), (R_j, \overline{R}_1)\} \cup \{(R_i, \overline{R}_i)\}_{i \ne 1,j}$.
  - $\overline{R}_h$, $2 \le h \le k+1$, is a match; the perfect matching is $\{(R_j, \overline{R}_h), (R_h, \overline{R}_j)\} \cup \{(R_i, \overline{R}_i)\}_{i \ne j,h}$.
- R_j, $k \le j \le k+1$, has k matches:
  - $\overline{R}_h$, $2 \le h \le k+1$, is a match; the perfect matching is $\{(R_j, \overline{R}_h), (R_h, \overline{R}_j)\} \cup \{(R_i, \overline{R}_i)\}_{i \ne j,h}$.

Finally, we turn to exemplify the non-emptiness of Ω_4. Consider the following table D alongside with a corresponding generalization, $g(D)$.

D	$g(D)$
$R_1 = a_1$	$\overline{R}_1 = \{a_1, b_1, \ldots, b_{k-1}\}$
$R_2 = a_2$	$\overline{R}_2 = \{a_1, \ldots, a_{k+1}\}$
$\vdots$	$\vdots$
$R_k = a_k$	$\overline{R}_k = \{a_1, \ldots, a_{k+1}\}$
$R_{k+1} = a_{k+1}$	$\overline{R}_{k+1} = \{a_2, \ldots, a_{k+1}\}$
$R_{k+2} = b_1$	$\overline{R}_{k+2} = \{b_1, \ldots, b_k\}$
$\vdots$	$\vdots$
$R_{2k+1} = b_k$	$\overline{R}_{2k+1} = \{b_1, \ldots, b_k\}$

It may be easily verified that $g(D)$ is a (k,k)-anonymization of D. However, it is not k-concealed since the first record in D, R_1, has only one match and that is the first record in $g(D)$, $\overline{R}_1$. Indeed, even though $\overline{R}_i$, $2 \leq i \leq k$, are all consistent with R_1, none of them is a match. Assume, on the contrary, that we try to extend one of the edges $(R_1, \overline{R}_i)$, $2 \leq i \leq k$, into a perfect matching. That is impossible since each of the k records $R_2, \ldots, R_{k+1}$ is consistent only with the k generalized records $\overline{R}_2, \ldots, \overline{R}_{k+1}$. But as $\overline{R}_i$ is already matched with R_1, that does not leave enough matches for $R_2, \ldots, R_{k+1}$.
$\square$

B The approximation guarantee of Algorithm 2

We define here two natural properties of the information loss measure Π — monotonicity and sub-additivity, and then proceed to prove that if the information loss measure satisfies those two properties, Algorithm 2 issues a $(k,1)$-anonymization that approximates the optimal one.

Definition 11. Let D be a database, let $g(D)$ be a generalization of D, and $g'(D)$ be a generalization of $g(D)$. Then a measure of loss of information, Π, is called monotone if $\Pi(D, g(D)) \leq \Pi(D, g'(D))$.

Definition 12. If for all subsets of records $S, T \subset A_1 \times \cdots \times A_r$ that have a non-empty intersection the following inequality holds,

$$d(S \cup T) \leq d(S) + d(T)\,,$$

the measure Π is called sub-additive.

Proposition 13. *Algorithm 2 produces a table $g(D) = \{\overline{R}_1, \ldots, \overline{R}_n\}$ that is a $(k,1)$-anonymization of D. Let Π be a measure of loss of information that respects monotonicity and sub-additivity. Then Algorithm 2 approximates optimal $(k,1)$-anonymity to within a factor of $k-1$, with respect to the cost measure Π.*

Proof. Each generalized record in $g(D)$ is consistent with at least k records in D, say, R_i, R_{i_1}, …, $R_{i_{k-1}}$, where the last $k-1$ records are those that were selected in Step 4 in the algorithm. Therefore, $g(D)$ is a $(k,1)$-anonymization of D.

Let $g'(D) = \{\overline{R}'_1, \ldots, \overline{R}'_n\}$ be an optimal $(k,1)$-anonymization of D. Since it is a $(k,1)$-anonymization of D, we infer that $\overline{R}'_i$ is consistent with, say, R_i, $R_{i'_1}$, …, $R_{i'_{k-1}}$. Without loss of generality, we may assume that the records $R_{i'_1}$, …, $R_{i'_{k-1}}$ are ordered in such a way that for every $1 \le \ell \le k-1$, $R_{i'_\ell} \notin \{R_{i_1}, R_{i_2}, \ldots, R_{i_{\ell-1}}\}$. We claim that under that assumption, the following inequality holds:

$$d(\{R_i, R_{i_1}, \ldots, R_{i_\ell}\}) \le d(\{R_i, R_{i_1}, \ldots, R_{i_{\ell-1}}\}) + d(\{R_i, R_{i'_\ell}\}) \, , \quad 1 \le \ell \le k-1 \, . \qquad (6)$$

Indeed, R_{i_ℓ} was selected in the ℓth application of Step 4 in the algorithm, after the records $R_{i_1}, \ldots, R_{i_{\ell-1}}$ have been already determined, as a record that minimized $d(\{R_i, R_{i_1},$ …, $R_{i_\ell}\})$. Since, by our assumption, $R_{i'_\ell}$ is not one of the first $\ell - 1$ records that were already selected at this stage, we infer that

$$d(\{R_i, R_{i_1}, \ldots, R_{i_{\ell-1}}, R_{i_\ell}\}) \le d(\{R_i, R_{i_1}, \ldots, R_{i_{\ell-1}}, R_{i'_\ell}\}) \, . \qquad (7)$$

By sub-additivity,

$$d(\{R_i, R_{i_1}, \ldots, R_{i_{\ell-1}}, R_{i'_\ell}\}) \le d(\{R_i, R_{i_1}, \ldots, R_{i_{\ell-1}}\}) + d(\{R_i, R_{i'_\ell}\}) \, . \qquad (8)$$

Inequality (6) now follows from (7) and (8).

Applying inequality (6) repeatedly for $\ell = k-1$ down to $\ell = 1$ we infer that

$$d(\{R_i, R_{i_1}, \ldots, R_{i_{k-1}}\}) \le \sum_{\ell=1}^{k-1} d(\{R_i, R_{i'_\ell}\}).$$

Monotonicity implies that $d(\{R_i, R_{i_1}, \ldots, R_{i_{k-1}}\}) \le (k-1) \cdot d(\{R_i, R_{i'_1}, \ldots, R_{i'_{k-1}}\})$. Since the left hand side in the last inequality equals $c(\overline{R})$ while the right hand side is bounded from above by $(k-1) \cdot c(\overline{R}'_i)$, we conclude that $c(\overline{R}_i) \le (k-1) \cdot c(\overline{R}'_i)$. Hence, the information-loss function $\Pi(D, g(D))$ may be bounded as follows:

$$\Pi(D, g(D)) = \frac{1}{n} \cdot \sum_{i=1}^{n} c(\overline{R}_i) \le (k-1) \cdot \frac{1}{n} \sum_{i=1}^{n} c(\overline{R}'_i) = (k-1) \cdot \Pi(D, g'(D)) \, .$$

□

We would like to stress that the approximation ratio guarantee in Proposition 13 is mainly of theoretical value because of two reasons. First, the proven approximation ratio, which is of the same order of magnitude as the approximation factor of the forest algorithm due to Aggarwal et al. in the context of k-anonymity [2], is quite large. The experimental results with both algorithms (see Section 9) provide a much better indication of the algorithms' performance in practice. The second reason is that Algorithm 2 must be coupled with the algorithms of the next steps (Algorithms 3 and 5) which are heuristical algorithms.

C Modifying k-anonymizations to meet the ℓ-diversity constraint

As noted in [32], any algorithm for k-anonymization may be enhanced so that it issues k-anonymized tables that are also ℓ-diverse. The idea is simple: The diversity of the union of two clusters of records is a convex combination of the diversities of the two clusters (this is true for all acceptable definitions of diversity). Hence, if there are clusters of records that violate ℓ-diversity, one can start unifying them until ℓ-diversity is met. Such a procedure, as explained in [32], will always stop successfully if the target diversity parameter ℓ is a legitimate one (namely, if the global diversity in D is at least ℓ). Therefore, in order to convert an algorithm that is designed to achieve only k-anonymity into one that achieves k-anonymity *and* ℓ-diversity, it is needed to post-process the output clustering by unifying clusters that violate ℓ-diversity until all clusters are ℓ-diverse. (Namely, as explained in Section 8, until in each cluster of records, the most frequent sensitive value appears in relative frequency which is no larger than $1/\ell$.)

Before describing the algorithm, we introduce the following notations:

Definition 14. Let $C = \{R_{i_1}, \dots, R_{i_{|C|}}\}$ be a cluster of records in D and let $C' := \{s_{i_1}, \dots, s_{i_{|C|}}\}$ be the private values of those records. Let f be the number of occurrences of the most frequent value in C'. Then the diversity of C is $\text{div}(C) := |C|/f$. Let $\gamma = \{C_1, \dots, C_b\}$ be a clustering of the records of the table D. Then its diversity is $\text{div}(\gamma) := \min_{1 \le i \le b} \text{div}(C_i)$.

Algorithm 8, that is described below, is a post-processing procedure that may be applied on top of any algorithm of k-anonymization. Its input is any clustering of the records of the table D, and a target diversity parameter $\ell \ge 1$. Its output is a coarser clustering in which all clusters are ℓ-diverse. (By a coarser clustering we mean that it is derived from the input clustering only by means of unifying clusters.) When Algorithm 8 is applied on clusterings issued by a k-anonymization algorithm, namely, clusterings in which all clusters are of size at least k, it will output a clustering in which all clusters are of size at least k, and, in addition, are ℓ-diverse.

First, the algorithm computes the diversities of all clusters in the input clustering γ. It selects the cluster C_m with minimal diversity. If that cluster is already ℓ-diverse, the clustering is ripe to be output. Otherwise, we look for the best cluster with which C_m can be unified. Once such a cluster is found, we unify it with C_m and then repeat the procedure until all clusters are ℓ-diverse.

The algorithm uses a cost function in order to decide about the most profitable unification. On one hand, unifying the least diverse cluster with another cluster brings us closer to meeting the ℓ-diversity requirement. On the other hand, unifying clusters increases the information loss. It is our goal to achieve a maximal gain towards meeting the ℓ-diversity requirement, but at the same time we wish to favor unifications that will incur smaller additions to the information loss. Hence, we define the cost function as a weighted average between an information cost and a diversity cost.

Let $\gamma = \{C_1, \dots, C_t\}$ be a clustering of the records in the table D. For any two clusters in γ, say C_i, C_j, we let γ_{C_i,C_j} denote the clustering that would be obtained from γ if C_i and C_j were unified (i.e., $\gamma_{C_i,C_j} = (\gamma \setminus \{C_i, C_j\}) \cup \{C_i \cup C_j\}$). Let $\text{cost}_I(C_i, C_j)$ denote the loss of information in case we decide to unify C_i and C_j; it equals the information loss of γ_{C_i,C_j} minus the information loss of γ. The diversity cost, $\text{cost}_D(C_i, C_j)$, is defined as the remaining gap between the diversity of the unified cluster $C_i \cup C_j$ and the target level ℓ:

$$\text{cost}_D(C_i, C_j) = \max\{\ell - \text{div}(C_i \cup C_j)\ ,\ 0\}\ . \tag{9}$$

Our goal is to minimize $\text{cost}_I(C_i, C_j)$ as well as $\text{cost}_D(C_i, C_j)$. To that end, we define the weighted cost function

$$\text{cost}(C_i, C_j) = w \cdot \text{cost}_I(C_i, C_j) + (1 - w) \cdot \text{cost}_D(C_i, C_j)\,, \tag{10}$$

where w is a weight between 0 and 1 that can be tuned experimentally. In our experiments we used $w = 0.15$, a value that was found to yield best results.

Algorithm 8 A post-processing algorithm to achieve ℓ-diverse anonymizations

Input: A clustering $\gamma = \{C_1, \ldots, C_t\}$ of the records in a table D; a target diversity parameter $\ell \geq 1$.

Output: A coarser clustering that respects ℓ-diversity.

```
1: Compute div($C_i$) for all $C_i \in \gamma$.
2: Let $C_m$ be the cluster with minimal diversity in $\gamma$.
3: if div($C_m$) $\geq \ell$ then
4:     Output $\gamma$ and stop.
5: end if
6: Compute cost($C_i, C_m$) for all $C_i \in \gamma \setminus \{C_m\}$.
7: Find the cluster $C_i \in \gamma \setminus \{C_m\}$ for which cost($C_i, C_m$) is minimal.
8: Remove $C_i$ and $C_m$ from $\gamma$ and add to $\gamma$ the cluster $C_i \cup C_m$.
9: Go to Step 2.
```