Extending Loose Associations to Multiple Fragments

Sabrina De Capitani di Vimercati[1], Sara Foresti[1], Sushil Jajodia[2], Giovanni Livraga[1], Stefano Paraboschi[3], and Pierangela Samarati[1]

[1] Università degli Studi di Milano, 26013 Crema, Italy
firstname.lastname@unimi.it
[2] George Mason University, Fairfax, VA 22030-4444, USA
jajodia@gmu.edu
[3] Università degli Studi di Bergamo, 24044 Dalmine, Italy
parabosc@unibg.it

Abstract. Data fragmentation has been proposed as a solution for protecting the confidentiality of sensitive associations when publishing data at external servers. To enrich the utility of the published fragments, a recent approach has put forward the idea of complementing them with *loose associations*, a sanitized form of the sensitive associations broken by fragmentation. The original proposal considers fragmentations composed of two fragments only, and supports the definition of a loose association between this pair of fragments. In this paper, we extend loose associations to multiple fragments. We first illustrate how the publication of multiple loose associations between pairs of fragments of a generic fragmentation can potentially expose sensitive associations. We then describe an approach for supporting the more general case of publishing a loose association among an arbitrary set of fragments.

Keywords: Loose associations, fragmentation, confidentiality constraints, privacy, data publishing

1 Introduction

The strong need for sharing and disseminating information that characterizes our global internetworked society raises a number of privacy concerns (e.g., [8, 12]). In fact, the vast amount of data collected and maintained in the digital infrastructure often includes sensitive information that must be adequately protected. There is then a clear trade off between the need of easily accessing, using, and distributing information, and the equally strong need of providing proper protection guarantees to sensitive information. Traditional solutions aimed at protecting data undergoing public or semi-public release are based on k-anonymity [17] and differential privacy [11], which protect respondents' identities and their sensitive information by releasing a sanitized version of the data. These solutions however are not applicable in scenarios characterized by the

need of releasing non-modified information. Recent solutions have proposed the use of *fragmentation* for protecting sensitive associations among data [1, 4, 5]. Intuitively, fragmentation protects sensitive associations among different pieces of data by storing them in different fragments that cannot be joined. On one hand, fragmentation improves data accessibility and query performance since it allows the storage of plaintext values at the server side. On the other hand the utility of the published data may be compromised since fragmentation breaks the associations existing in the original data collection. This problem has been addressed by observing that often it may be sufficient to guarantee that sensitive associations cannot be precisely reconstructed (i.e., they can be reconstructed with a minimum degree k of uncertainty). For instance, if the association between patients' names and the disease they suffer from is sensitive, it would be sufficient to guarantee that a recipient cannot associate each patient with less than k possible diseases. In these scenarios, fragments can be complemented with *loose associations* [9], which permit to partially reconstruct the association between sub-tuples in fragments, while not precisely disclosing the association among attribute values that are considered sensitive. Loose associations partition the tuples in fragments in groups and release the associations between sub-tuples in fragments at the granularity of group (instead of the precise tuple-level association). Loose associations can then be used for evaluating aggregate queries, with limited errors in the result, and for data mining. The existing approach operates under the assumption that a fragmentation includes two fragments only, and produces a single loose association between this pair of fragments. A fragmentation may however include an arbitrary number of fragments, and the definition of a loose associations may then consider the presence of multiple fragments. A naive solution would publish multiple loose associations, one for each pair of fragments involved in associations that need to be loosely released. Such an approach unfortunately opens the door to privacy breaches since associations that go beyond the two fragments are not considered, and could then be exposed (i.e., a recipient could be able to reconstruct them).

In this paper, we aim at overcoming such a limitation, proposing a solution for the definition of loose associations among arbitrary sets of fragments. The remainder of the paper is organized as follows. Section 2 introduces the basic concepts. Section 3 illustrates the privacy risks caused by the release of multiple loose associations. Section 4 presents our definition of loose association among an arbitrary set of fragments. Section 5 describes the heterogeneity properties ensuring that a loose association satisfies a given privacy degree. Section 6 discusses the advantages and some interesting properties enjoyed by our novel formulation. Section 7 discusses related work. Finally, Section 8 concludes the paper.

2 Basic Concepts

We consider a scenario where a data owner publishes a set $\mathcal{F} = \{F_1, \ldots, F_n\}$ of fragments of a private relation S. Sensitive associations among attributes in S

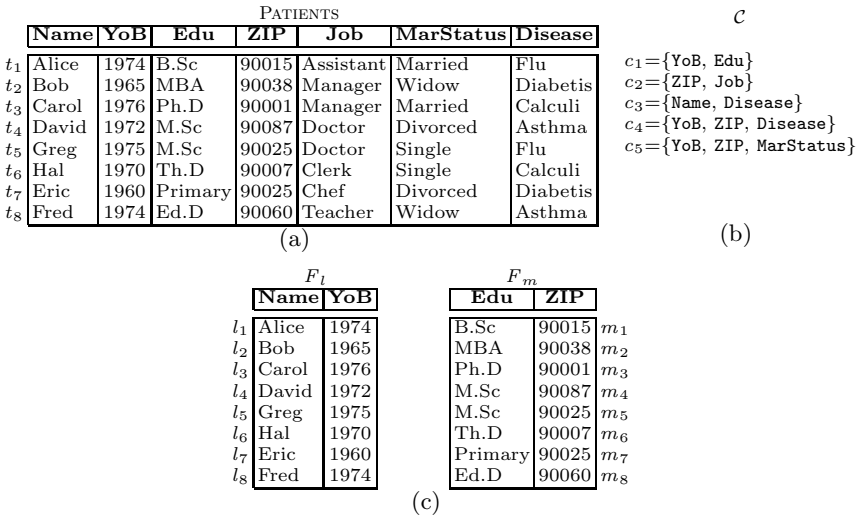

Fig. 1. An example of relation (a), a set $\mathcal{C}$ of confidentiality constraints over it (b), and a minimal fragmentation that satisfies the constraints in $\mathcal{C}$ (c)

that should not be revealed by the release of $\mathcal{F}$ are modeled through *confidentiality constraints* [1].

Definition 1 (Confidentiality constraint). *Given a relation schema S, a* confidentiality constraint *c over S is a subset of the attributes in S.*

Figure 1(b) illustrates a set $\mathcal{C}$ of confidentiality constraints defined over relation PATIENTS in Figure 1(a). A fragmentation can be published only if it satisfies all the confidentiality constraints, that is, only if it does not disclose sensitive associations, neither directly in a single fragment (i.e., $\forall F \in \mathcal{F}, \forall c \in \mathcal{C} : c \not\subseteq F$), nor indirectly by joining fragments (i.e., fragments are disjoint, $\forall F_i, F_j \in \mathcal{F}, i \neq j$: $F_i \cap F_j = \emptyset$). In our discussion, we assume the released fragmentation to be *minimal*, meaning that merging fragments in $\mathcal{F}$ would violate at least a confidentiality constraint. This is in line with the idea that the data owner does not fragment relation S more than necessary. Figure 1(c) illustrates an example of minimal fragmentation of relation PATIENTS in Figure 1(a), which satisfies the constraints in Figure 1(b).

To mitigate information loss caused by the fact that fragmentation breaks the associations in the original relation, fragments can be complemented with *loose associations*, introduced in [9] for fragmentations composed of a single pair of fragments. To this aim, tuples in the two fragments are partitioned in groups, and information on the associations between tuples in fragments is released at the group (in contrast to tuple) level.

Given a fragmentation $\mathcal{F}=\{F_l, F_m\}$ and its instance $\{f_l, f_m\}$, tuples in f_l and f_m are first independently partitioned in groups of size at least k_l and

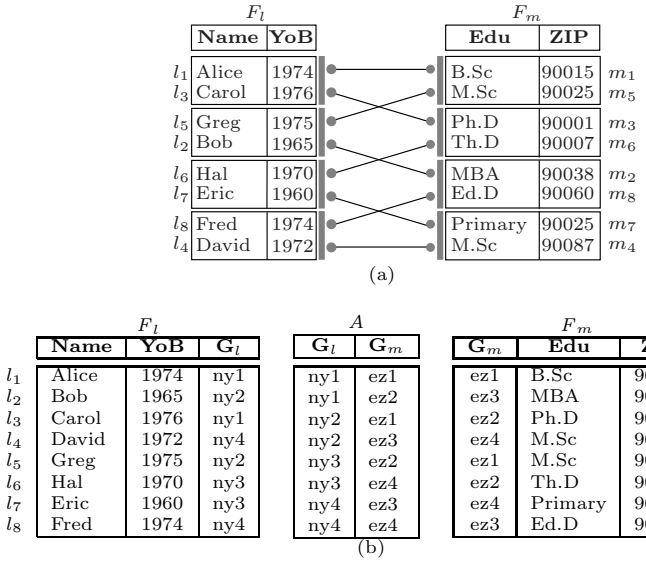

Fig. 2. Graphical representation (a) and corresponding relations (b) of a 4-loose association between fragments F_l and F_m in Figure 1(c)

k_m, respectively. To this aim, the data owner defines a *k-grouping* function for each of the two fragments. A k-grouping function partitions tuples in a fragment instance f in groups of size at least k, by associating a group identifier with each tuple in f [9].

Definition 2 (k-Grouping). *Given a fragment F_i, its instance f_i, and a set* GID_i *of group identifiers, a* k-grouping *function over f_i is a surjective function* $\mathcal{G}_i{:}f_i{\rightarrow}\mathsf{GID}_i$ *such that* $\forall g_i \in \mathsf{GID}_i :\mid \mathcal{G}_i^{-1}(g_i) \mid \geq k$.

Notation (k_l,k_m)-grouping denotes a k_l-grouping over f_l and a k_m-grouping over f_m (note that k_l may be different from k_m). Once each tuple in f_l and in f_m is associated with a group identifier, the group-level relationships between tuples in f_l and in f_m are represented by an additional relation A. For each tuple t in the original relation, relation A includes a tuple containing the group where $t[F_l]$ appears in f_l and the group where $t[F_m]$ appears in f_m. For instance, Figure 2(a) represents a partition in groups of size k_l=k_m=2 of the tuples in fragments f_l and f_m in Figure 1(c). For simplicity, given a tuple t in the original relation, we denote with l (m, resp.) the sub-tuple $t[F_l]$ ($t[F_m]$, resp.) in fragment f_l (f_m, resp.). The edges connecting grey dots, which correspond to tuples in groups, represent the group-level associations between tuples in the two fragments implied by relation PATIENTS in Figure 1(a). Figure 2(b) illustrates relation A and fragments F_l and F_m enriched with an attribute ($\mathtt{G}_l$ and $\mathtt{G}_m$, resp.) reporting the identifier of the group to which each tuple belongs.

The partitioning of the tuples in the two fragments should be carefully designed to guarantee that sensitive associations cannot be reconstructed exploiting A. Intuitively, a loose association between a pair of fragments $\{F_l, F_m\}$ enjoys a degree k of protection (referred to as k-looseness) if every tuple in A indistinguishably corresponds to at least k distinct associations among tuples in f_l and f_m with different values for the attributes involved in each confidentiality constraint c such that $c \subseteq F_l \cup F_m$. In fact, the release of a loose association between F_l and F_m only puts at risk constraints whose attributes are all represented by the two fragments. For instance, the first tuple in table A in Figure 2(b) corresponds to four possible tuples (i.e., $\langle l_1, m_1 \rangle$, $\langle l_1, m_5 \rangle$, $\langle l_3, m_1 \rangle$, $\langle l_3, m_5 \rangle$), all with different values for attributes `YoB` and `Edu` composing constraint c_1, which is the only constraint completely covered by F_l and F_m. In other words, the release of F_l, F_m, and A satisfies *k-looseness* for each $k \leq k_l \cdot k_m$, if for each group g_l in f_l (g_m in f_m, resp.), the union of the tuples in all the groups with which g_l (g_m, resp.) is associated in A is a set of at least k tuples with different values for the attributes in each constraint $c \subseteq F_l \cup F_m$ [9]. For instance, the association in Figure 2 satisfies k-looseness for any $k \leq 4$.

3 Problem and Motivating Example

Although effective for publishing loose associations between pairs of fragments, the proposal in [9] cannot be directly applied to the release of multiple loose associations between different pairs of fragments, since they might disclose sensitive associations. To illustrate the problem, consider a fragmentation $\mathcal{F}$ composed of 3 fragments, say F_l, F_m, and F_r. A straightforward approach to release group-level associations among these fragments consists in releasing two distinct loose associations between two pairs of fragments in $\mathcal{F}$ (e.g., one between F_l and F_m, and one between F_m and F_r). For instance, consider a fragmentation of relation Patients in Figure 1(a) that satisfies the constraints in Figure 1(b), composed of 3 fragments F_l=$\{$`Name`, `YoB`$\}$, F_m=$\{$`Edu`, `ZIP`$\}$, and F_r=$\{$`Job`, `MarStatus`, `Disease`$\}$. Figure 3 illustrates a 4-loose association between F_l and F_m (A_{lm}), and a 4-loose association between F_m and F_r (A_{mr}) (note that tuples in f_m are partitioned according to two different grouping functions, one for each loose association).

Such an approach clearly releases useful information on the associations between the tuples in F_l and F_m, and between the tuples in F_m and F_r. The loose associations between F_l and F_m, and between F_m and F_r imply however an *induced association* between F_l and F_r: F_l can be loosely joined with F_m, which in turn can be loosely joined with F_r. Therefore, each tuple in f_l is associated with a group of tuples in f_m, each of which is in turn associated with a group of tuples in f_r. As an example, tuple l_7 in fragment f_l in Figure 3 is associated with tuples m_3, m_4, m_6, and m_7 in fragment f_m. In turn, m_3 and m_6 are associated with r_1, r_3, r_5, and r_6 in f_r. Tuples m_4 and m_7 are instead associated with r_2, r_4, r_7, and r_8 in f_r. Therefore, l_7 is possibly associated with any tuple in f_r. The induced association between F_l and F_r might then seem to enjoy a

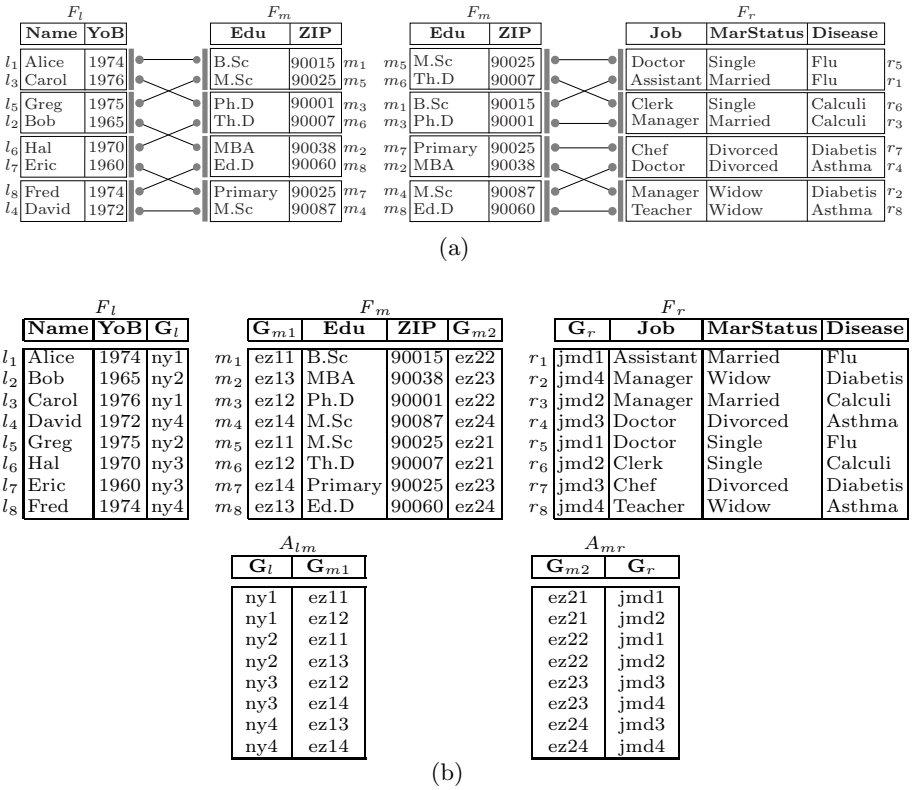

Fig. 3. Graphical representation (a) and corresponding relations (b) of a 4-loose association A_{lm} between F_l and F_m, and a 4-loose association A_{mr} between F_m and F_r, with F_l, F_m, and F_r three fragments of relation PATIENTS in Figure 1(a)

protection degree equal to (or even greater than) those enjoyed by A_{lm} and A_{mr}. However, publishing loose associations A_{lm} and A_{mr} guarantees that sensitive associations involving only attributes in F_l and F_m, and only attributes in F_m and F_r are protected. It does not provide any guarantee on the protection of sensitive associations involving attributes stored in F_l and in F_r, which are possibly exposed by the induced association. This is due to the fact that the loose association between F_l and F_m requires tuples in f_l (f_m, resp.) associated with each group in f_m (f_l, resp.) to have different values for the attributes appearing in constraints $c \subseteq F_l \cup F_m$ (c_1, in our example). Analogously, the loose association between F_m and F_r requires tuples in f_m (f_r, resp.) associated with each group in f_r (f_m, resp.) to have different values for the attributes appearing in constraints $c \subseteq F_m \cup F_r$ (c_2, in our example). Constraints $c \subseteq F_l \cup F_m \cup F_r$ such that $c \cap F_l \neq \emptyset$ and $c \cap F_r \neq \emptyset$ (c_3, c_4, c_5, in our example) are instead ignored. To illustrate, the release of the fragments and loose associations in Figure 3 exposes

the sensitive association between attributes `Name` and `Disease`, violating constraint c_3 in Figure 1(b). In fact, tuple l_1 in f_l is associated with tuples m_1, m_3, m_5, and m_6 in f_m. In turn, m_1, m_3, m_5, and m_6 in f_m are all associated with tuples r_1, r_3, r_5, and r_6 in f_r. Thus, the observation of A_{lm} and A_{mr} reveals that l_1 is associated, in the original relation, with one among r_1, r_3, r_5, and r_6, but r_1[`Disease`]=r_5[`Disease`]=Flu and r_3[`Disease`]=r_7[`Disease`]=Calculi. Therefore, either association ⟨Alice,Flu⟩ or association ⟨Alice,Calculi⟩ belongs to relation PATIENTS with the same probability. The degree of protection for constraint c_3 offered by the release of the two loose associations in Figure 3 is then 2 (and not 4 as for constraints c_1 and c_2). Note that the release of arbitrary loose associations may completely expose sensitive associations. For instance, assume that r_3[`Disease`]=r_6[`Disease`]=Flu. The released associations would still be 4-loose, but they reveal that Alice suffers from Flu.

The privacy breach described above represents a serious issue for the data owner since it exposes sensitive associations that she is not explicitly publishing. She could then be unaware of the fact that the released fragments and loose associations expose sensitive associations. In the remainder of this paper, we illustrate our proposal for counteracting such a privacy problem. Our intuition is to define a single loose association encompassing all the fragments among which the data owner needs to publish group-level associations. In this way, we aim at defining one loose association only that takes into consideration *all* the confidentiality constraints among attributes stored by the released fragments. Intuitively, since all the published fragments are involved in the same loose association, publishing this association does not imply the disclosure of induced associations that can be exploited by malicious recipients to precisely reconstruct sensitive associations. As we will detail in the following sections, starting from this loose association, the data owner may then choose to either release it as a whole, or use it to build an arbitrary set of loose associations, with the guarantee that no sensitive association be improperly exposed.

4 Loose Associations

Given a fragmentation $\mathcal{F}$ of a relation S and a set $\mathcal{C}$ of confidentiality constraints over S, we define a loose association among the fragments in $\mathcal{F}$ (note that our approach also permits to define a loose association among an arbitrary subset of fragments in $\mathcal{F}$). For the sake of readability, we refer the discussion to a fragmentation $\mathcal{F}$=$\{F_l, F_m, F_r\}$ composed of 3 fragments, while definitions are formulated on fragmentations composed of an arbitrary number of fragments. In line with previous works on fragmentation [1, 4, 9], we assume that data recipients do not possess any additional knowledge besides released fragments, loose associations, and confidentiality constraints defined by the data owner.

The first step necessary for the definition of a loose association among the fragments in $\mathcal{F}$ is the identification of the subset of confidentiality constraints in $\mathcal{C}$ that are *relevant* for $\mathcal{F}$. A constraint is relevant for a set $\{F_1, \ldots, F_n\}$ of fragments if it includes only attributes represented by the fragments in $\{F_1, \ldots, F_n\}$.

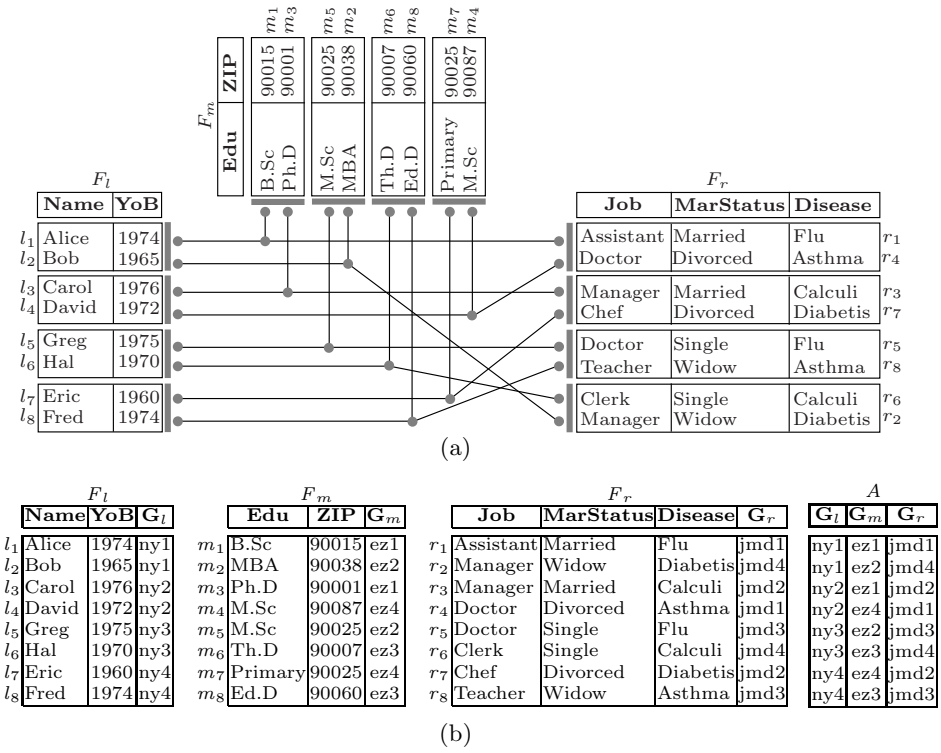

Fig. 4. Graphical representation (a) and corresponding relations (b) of a 4-loose association among three fragments F_l, F_m, and F_r of relation PATIENTS in Figure 1(a)

Indeed, any other constraint cannot be violated by the release of a loose association among fragments in $\{F_1, \ldots, F_n\}$.

Definition 3 (Relevant constraints). *Given a set $T = \{F_1, \ldots, F_n\}$ of fragments and a set $\mathcal{C}$ of confidentiality constraints, the set $\mathcal{C}_T$ of* relevant constraints *for T is defined as $\mathcal{C}_T = \{c \in \mathcal{C} : c \subseteq F_1 \cup \ldots \cup F_n\}$.*

For instance, the only constraint in Figure 1(b) relevant for the set of fragments in Figure 1(c) is c_1 as it is the only constraint whose attributes belong to the set {`Name`, `YoB`, `Edu`, `ZIP`}.

Given a fragmentation $\mathcal{F}=\{F_1, \ldots, F_n\}$, the tuples in each fragment are partitioned according to different grouping functions, which may adopt different protection parameters (thus generating groups of different size). A $(k_1, \ldots, k_n)$-grouping is a set $\{\mathcal{G}_1, \ldots, \mathcal{G}_n\}$ of grouping functions defined over fragments in $\{f_1, \ldots, f_n\}$ (i.e., a set of k_i-groupings over f_i, i=1, …, n). As an example, Figure 4 illustrates a (2,2,2)-grouping involving fragments F_l={`Name`, `YoB`},

F_m={Edu, ZIP} and F_r={Job, MarStatus, Disease} of relation PATIENTS in Figure 1(a), and the corresponding group association. It is easy to see that tuple t_1 in relation PATIENTS is represented in fragments F_l, F_m, and F_r by tuples l_1, m_1, and r_1, respectively. The association among l_1, m_1, and r_1 is represented by tuple ⟨ny1,ez1,jmd1⟩ in A, which defines an association among the groups to which l_1, m_1, and r_1 belong.

A group association A can be safely released only if it cannot be exploited to reconstruct, totally or in part, sensitive associations among the released fragments. A (k_l,k_m,k_r)-grouping guarantees that each tuple in A corresponds to $k_l \cdot k_m \cdot k_r$ different associations among tuples in f_l, f_m, and f_r. However, some tuples represented by these $k_l \cdot k_m \cdot k_r$ associations might have the same values for the attributes in a relevant constraint, thus reducing in practice the protection degree enjoyed by the published group association. To guarantee that a group association A does not expose relevant confidentiality constraints, each tuple in A must refer to k distinct associations among sub-tuples in fragments that do not have the same values for the attributes in relevant constraints. A group association satisfying this property is said to be *k-loose*. To compare the values assumed in fragments by the attributes in relevant constraints, we formally introduce the *alike* relationship between tuples as follows.

Definition 4 (Alike). *Given a fragmentation $\mathcal{F}=\{F_1,\ldots,F_n\}$ with its instance $\{f_1,\ldots,f_n\}$, and the set $\mathcal{C}_\mathcal{F}$ of confidentiality constraints relevant for $\mathcal{F}$, $t_i,t_j \in f_z$, $z=1,\ldots,n$, are said to be* alike *with respect to a constraint $c \in \mathcal{C}_\mathcal{F}$, denoted $t_i \simeq_c t_j$ iff $c \cap F_z \neq \emptyset \wedge t_i[c \cap F_z] = t_j[c \cap F_z]$. Two tuples are said to be* alike *with respect to a set $\mathcal{C}_\mathcal{F}$ of relevant constraints, denoted $t_i \simeq_{\mathcal{C}_\mathcal{F}} t_j$, if they are alike with respect to at least one constraint $c \in \mathcal{C}_\mathcal{F}$.*

Definition 4 states that given a fragmentation $\mathcal{F}$, two tuples in a fragment instance f_i are alike if they have the same values for the attributes in a constraint relevant for $\mathcal{F}$. For instance, with reference to the (2,2,2)-grouping in Figure 4, $r_4 \simeq_{c_3} r_8$ since r_4[Disease]=r_8[Disease]=Asthma. Since we are interested in evaluating the alike relationship w.r.t. the set $\mathcal{C}_\mathcal{F}$ of relevant constraints, in the following we omit the subscript of the alike relationship whenever clear from the context (i.e., we write $t_i \simeq t_j$ instead of $t_i \simeq_{\mathcal{C}_\mathcal{F}} t_j$). The alike relationship guides the definition of k-loose group associations among arbitrary sets of fragments, as formally defined in the following.

Definition 5 (k-Looseness). *Given a fragmentation $\mathcal{F}=\{F_1,\ldots,F_n\}$ with its instance $\{f_1,\ldots,f_n\}$, the set $\mathcal{C}_\mathcal{F}$ of confidentiality constraint relevant for $\mathcal{F}$, and a group association A over $\{f_1,\ldots,f_n\}$, A is said to be* k-loose *w.r.t. $\mathcal{C}_\mathcal{F}$ iff $\forall c \in \mathcal{C}_\mathcal{F}$, $\forall F_i \in \mathcal{F} : c \cap F_i \neq \emptyset$ and $\forall g_i \in \mathsf{GID}_i$, $\exists F_j \in \mathcal{F} : c \cap F_j \neq \emptyset$ that satisfies the following condition: let $T = \bigcup_z \{\mathcal{G}_j^{-1}(g_z) \mid (g_i,g_z) \in A[G_i,G_j]\} \Longrightarrow |\,T\,| \geq k$, and $\forall t_x,t_y \in T$, $x \neq y$, $t_x \not\simeq_c t_y$.*

k-Looseness guarantees that sensitive associations represented by relevant constraints cannot be reconstructed with confidence higher than $1/k$. According to the definition above, a group association A is k-loose if each tuple in A

corresponds to k possible tuples in the original relation that are not alike w.r.t. any relevant constraint. Note that, however, there are cases in which the alike requirement can be relaxed. In fact, whenever a value v in the domain of an attribute is considered not sensitive (e.g., because it characterizes the majority of the tuples in the original relation), the alike relationship may consider such a value as *neutral*. In this case, even if t_i[Attr]=t_j[Attr]=v, $t_i \not\approx t_j$.

The definition of k-looseness translates into the satisfaction of a different condition depending on whether the considered constraint involves two (like in [9]) or more fragments.

- *Constraints between two fragments.* k-Looseness requires that, for each group g_l in f_l, the union of the tuples in all the groups g_m in f_m with which g_l is associated is a set including at least k tuples that are not alike w.r.t. c (and viceversa). With reference to the example in Figure 4, c_1 cannot be reconstructed since each group in f_l is associated with two different groups in f_m including tuples that do not contain duplicates for Edu and viceversa.
- *Constraints among more than two fragments.* k-Looseness requires to break the association between at least two of the fragments storing attributes in c to guarantee that the sensitive association represented by c cannot be reconstructed. We then need to guarantee that, for each group g_l in f_l, the union of the tuples in all the groups g_m in f_m with which g_l is associated *or* the union of the tuples in all the groups g_r in f_r with which g_l is associated is a set of at least k tuples that are not alike w.r.t. c. Clearly, this property must hold also for each group g_m in f_m and for each group g_r in f_r. For instance, consider the fragments and group association in Figure 4 and constraint c_5 over them. Sensitive associations among YoB, ZIP, and MarStatus cannot be reconstructed even if group ny2 in f_l is associated with groups jmd1 and jmd2 in f_r whose tuples have the same values for attribute MarStatus. In fact, group ny2 is associated with groups ez1 and ez4 in f_m, which do not include tuples that are alike w.r.t. c_5 (i.e., tuples in ez1 and ez4 have all different values for ZIP).

This definition of k-looseness implies that the release of a (k_l,k_m,k_r)-grouping induces a k-loose association with k=min($k_l \cdot k_m$, $k_m \cdot k_r$, $k_l \cdot k_r$). In fact, the constraints relevant for $\{F_l,F_m\}$ ($\{F_m,F_r\}$ and $\{F_l,F_r\}$, resp.) enjoy a protection degree k_{lm}=$k_l \cdot k_m$ (k_{mr}=$k_m \cdot k_r$ and k_{lr}=$k_l \cdot k_r$, resp.). Constraints relevant for $\{F_l,F_m,F_r\}$ enjoy the minimum protection degree among k_{lm}, k_{mr}, and k_{lr} since, as illustrated above, it is not required that all the associations among the attributes in the constraints be broken. Figure 4(b) illustrates the 4-loose association induced by the (2,2,2)-grouping in Figure 4(a). This association guarantees the same protection degree k=k_{lm}=k_{mr}=k_{lr}=4 to each pair of fragments (and then also to $\mathcal{F}$).

5 Heterogeneity Properties

In this section, we enhance and extend the heterogeneity properties (i.e., group, association, and deep heterogeneity), originally proposed to guarantee that a

group association between two fragments is k-loose, to provide the same guarantee to group associations defined on an arbitrary number of fragments.

Group Heterogeneity. This property guarantees that groups do not include tuples with the same values for the attributes in relevant constraints. In this way, the minimum size k_i of the groups in fragment F_i, $i = 1, \ldots, n$, reflects the minimum number of different values in the group for each subset of attributes that appear together in a relevant constraint.

Property 1 (Group heterogeneity). Given a fragmentation $\mathcal{F} = \{F_1, \ldots, F_n\}$ with its instance $\{f_1, \ldots, f_n\}$, and the set $\mathcal{C}_\mathcal{F}$ of constraints relevant for $\mathcal{F}$, grouping functions $\mathcal{G}_i$ over f_i, $i = 1, \ldots, n$, satisfy *group heterogeneity* iff $\forall f_i \in \{f_1, \ldots, f_n\}$, $\forall t_z, t_w \in f_i$: $t_z \simeq t_w \implies \mathcal{G}_i(t_z) \neq \mathcal{G}_i(t_w)$.

The definition of this property is similar to the one operating on two fragments, as it is local to the tuples in each fragment. It however operates on a different set of constraints, that is, the set of constraints relevant for $\mathcal{F}$. For instance, in Figure 4 the grouping functions defined for the three fragments satisfy group heterogeneity for $\mathcal{C}_\mathcal{F} = \{c_1, \ldots, c_5\}$. On the contrary, the groupings for the three fragments in Figure 3 do not satisfy group heterogeneity for $\mathcal{F} = \{F_l, F_m, F_r\}$ since, for example, $r_1 \simeq_{c_3} r_5$ and they belong to the same group. However, these groupings satisfy group heterogeneity for $\mathcal{F}_1 = \{F_l, F_m\}$ (where c_1 is the only relevant constraint) and for $\mathcal{F}_2 = \{F_m, F_r\}$ (where c_2 is the only relevant constraint).

Association Heterogeneity. For loose associations between two fragments, this property requires that A cannot have duplicates. This simple condition is however not sufficient in our (more general) scenario. In fact, association heterogeneity aims at guaranteeing that, for each constraint c in $\mathcal{C}_\mathcal{F}$, each group in f_i is associated with at least k_i different groups in *at least* one of the fragments storing attributes in c (i.e., groups in f_j such that $c \cap F_j \neq \emptyset$). If a group in f_i is associated with one group in f_j only, it is easier for an observer to reconstruct the correct associations among the tuples in these two groups (and therefore to violate constraints). This condition implies that A cannot have two tuples with the same group identifier for all the fragments storing attributes composing a constraint (for constraints involving more than two fragments, it is sufficient that one of the values in the tuple be different).

Since we consider minimal fragmentations, there exists at least one relevant constraint for each pair of fragments in $\mathcal{F}$ (i.e., $\forall \{f_i, f_j\} \subseteq \mathcal{F}$, $i \neq j$, $\exists c \in \mathcal{C}$ s.t. $c \subseteq F_i \cup F_j$, Theorem A.2 in [9]). Therefore, a group association A satisfies association heterogeneity if it does not have two tuples with the same group identifier for any pair of fragments in $\mathcal{F}$.

Property 2 (Association heterogeneity). A group association A satisfies *association heterogeneity* iff $\forall (g_{i_1}, \ldots, g_{i_n}), (g_{j_1}, \ldots, g_{j_n}) \in A$: $i_z = j_z \implies i_w \neq j_w$, $w = 1, \ldots, n$ and $w \neq z$.

Intuitively, association heterogeneity requires that the projection over A of any subset of two attributes does not contain duplicate tuples. It is immediate to see that the group association in Figure 4 satisfies association heterogeneity.

Deep Heterogeneity. This property guarantees that a group in f_i cannot be associated with different groups in f_j including duplicated values for the attributes in a relevant constraint $c \subseteq F_i \cup F_j$. The groups in f_j with which a group in f_i is associated may be composed of tuples with exactly the same values for the attributes in c, limiting the protection offered by the loose association. For instance, groups jmd1 and jmd3 in Figure 4 have the same values for attribute `Disease` (i.e., Flu and Asthma). Therefore, a group in f_l cannot be associated with both jmd1 and jmd3 because of constraint c_3 (otherwise, the association between F_l and F_r would be 2-loose instead of 4-loose).

Deep heterogeneity imposes diversity by looking at the values behind the groups. The definition of deep heterogeneity over pairs of fragments requires that the groups in fragment f_i with which a group in f_j is associated in A do not contain alike tuples. A straightforward approach to extend deep heterogeneity would require diversity over all the fragments storing the attributes composing the constraint. In other words, considering a constraint c composed of attributes stored in fragments $\{F_1, \ldots, F_n\}$, all the groups in each fragment f_i $(i = 1, \ldots, n)$ with which a group in f_j $(j = 1, \ldots, n,\ i \neq j)$ is associated in A should not contain tuples that are alike w.r.t. c. This condition is more restrictive than necessary to define a k-loose association. In fact, it is sufficient, for each fragment F_j, to break the association with *one* of the fragments F_i $(i = 1, \ldots, n,\ i \neq j)$ storing the attributes in c. For instance, with reference to the example in Figure 4, it is sufficient that each group in f_l be associated with groups of non-alike tuples in either f_m or f_r to guarantee that the sensitive association modelled by c_5 is not exposed.

Property 3 (Deep heterogeneity). Given a fragmentation $\mathcal{F} = \{F_1, \ldots, F_n\}$ with its instance $\{f_1, \ldots, f_n\}$, and the set $\mathcal{C}_\mathcal{F}$ of constraints relevant for $\mathcal{F}$, a group association A over $\mathcal{F}$ satisfies *deep heterogeneity* iff $\forall c \in \mathcal{C}_\mathcal{F}$; $\forall F_z \in \mathcal{F}$, $F_z \cap c \neq \emptyset$; $\forall\ (g_{i_1}, g_{i_2}\ \ldots\ g_{i_n}), (g_{j_1}, g_{j_2} \ldots\ g_{j_n}) \in A$ the following condition is satisfied:

$$i_w = j_w \implies \bigvee_{l=1,\ldots,n,\ l \neq w} \nexists t_x, t_y\colon t_x \in \mathcal{G}_l^{-1}(g_{i_l}),\ t_y \in \mathcal{G}_l^{-1}(g_{j_l}),\ t_x \simeq_c t_y.$$

Given a constraint c whose attributes appear in fragments $\{F_{i_1}, \ldots, F_{i_j}\}$, deep heterogeneity is satisfied w.r.t. c if the set of tuples in the groups $\{g_{x_1}, \ldots, g_{x_w}\}$ in f_{i_x} with which a group g_y in f_{i_y} is associated are not alike w.r.t. c, for at least one fragment f_{i_x}, $x = 1, \ldots, j$ and $x \neq y$. This property must be true for all the groups in each fragment F_{i_x}, $x = 1, \ldots, j$. This guarantees that, for each constraint, no association can be precisely reconstructed by an observer. An example of group association that satisfies deep heterogeneity is illustrated in Figure 4. Note that deep heterogeneity is satisfied even though group ny2 in f_l is associated with groups jmd1 and jmd2 in f_r, which include tuples $r_1 \simeq_{c_5} r_3$ and $r_4 \simeq_{c_5} r_7$. In fact, group ny2 is also associated with groups

ez1 and ez4 in f_m that do not include tuples that are alike w.r.t. c_5 (i.e., with the same value for ZIP).

If the three properties above are satisfied by a $(k_1, \ldots, k_n)$-grouping and its induced group association, the group association is k-loose with $k \leq min(k_i \cdot k_j)$ $\forall i, j = 1, \ldots, n$, $i \neq j$, as stated by the following theorem (the proof has been omitted for space constraints).

Theorem 1. *Given a fragmentation $\mathcal{F} = \{F_1, \ldots, F_n\}$ with its instance $\{f_1, \ldots, f_n\}$, the set $\mathcal{C}_\mathcal{F}$ of constraints relevant for $\mathcal{F}$, and a $(k_1, \ldots, k_n)$-grouping that satisfies Properties 1, 2, and 3, the group association A induced by the $(k_1, \ldots, k_n)$-grouping is k-loose w.r.t. $\mathcal{C}_\mathcal{F}$ (Definition 5) for each $k \leq \min(k_i \cdot k_j)$, with $i, j = 1, \ldots, n$, $i \neq j$.*

As a consequence of the above theorem, the protection degree that a $(k_1, \ldots, k_n)$-grouping that satisfies Properties 1, 2, and 3 offers may be different for each confidentiality constraint c in $\mathcal{C}_\mathcal{F}$. Indeed, the protection degree for a constraint c is $min(k_i \cdot k_j)$, where $F_i, F_j \in \{F \in \mathcal{F} : F \cap c \neq \emptyset\}$.

In this paper, for space constraints, we do not discuss how to compute a k-loose association among an arbitrary set of fragments. We note however that the solution in [9] can be extended to our scenario, by properly modifying the enforcement of the above heterogeneity properties.

6 Discussion

The consideration of all the constraints relevant for the fragments involved in the loose association guarantees that no constraint can be violated. Thus, our loose association defined over an arbitrary set of fragments does not suffer from the confidentiality breach illustrated in Section 3, mainly caused by the fact that confidentiality constraints relevant for the fragments involved in induced associations are ignored. As an example, with reference to the 4-loose association in Figure 4, each tuple in A corresponds to four different associations of (different) values for attributes Name and Disease. This guarantees that constraint c_3 is satisfied, while it is violated by the example in Figure 3.

The release of a k-loose association among a set $\mathcal{F}$ of fragments is equivalent to the release of $2^n - n$, with $n = |\mathcal{F}|$, k-loose associations (one for each subset of fragments in $\mathcal{F}$). Indeed, the projection over a subset of attributes in A represents a k-loose association for the fragments corresponding to the projected attributes.

Observation 1 *Given a fragmentation $\mathcal{F}=\{F_1, \ldots, F_n\}$, a subset $\{F_i, \ldots, F_j\}$ of $\mathcal{F}$, and a k-loose association $A(\mathtt{G}_1, \ldots, \mathtt{G}_n)$ over $\mathcal{F}$, group association $A'(\mathtt{G}_i, \ldots, \mathtt{G}_j) = \pi_{(\mathtt{G}_i, \ldots, \mathtt{G}_j)}(A)$ is a k-loose association over $\{F_i, \ldots, F_j\}$.*

For instance, with reference to the 4-loose association in Figure 4, the projection of attributes $\mathtt{G}_l$, $\mathtt{G}_m$ in A is a 4-loose association between F_l and F_m.

Since a k-loose association defined over a set $\mathcal{F}$ of fragments guarantees that sensitive associations represented by constraints in $\mathcal{C}_\mathcal{F}$ are properly protected, the release of multiple loose associations among arbitrary (and possibly

overlapping) subsets of fragments in $\mathcal{F}$ provides the data owner with the same protection guarantee. The data owner can therefore decide to release either one loose association A encompassing the associations among the fragments in $\mathcal{F}$, or a subset of loose associations defined among arbitrary subsets of fragments in $\mathcal{F}$ by projecting the corresponding attributes from A.

Observation 2 *Given a fragmentation $\mathcal{F}$=$\{F_1,\ldots,F_n\}$ and a k-loose association $A(\mathtt{G}_1,\ldots,\mathtt{G}_n)$ over it, the release of an arbitrary set of k-loose associations $\{A_1(\mathtt{G}_h,\ldots,\mathtt{G}_i),\ldots,A_m(\mathtt{G}_j,\ldots,\mathtt{G}_k)\}$ with $\{\mathtt{G}_h,\ldots,\mathtt{G}_k\}\subseteq\{\mathtt{G}_1,\ldots,\mathtt{G}_n\}$ provides the same protection guarantee as the release of A.*

For instance, with reference to our examples above, aiming at releasing two distinct 4-loose associations, the data owner can release the 4-loose associations obtained projecting $\langle \mathtt{G}_l,\mathtt{G}_m\rangle$ and $\langle \mathtt{G}_m,\mathtt{G}_r\rangle$ from the 4-loose association in Figure 4. This solution does not suffer from the privacy breach illustrated in Section 3, while providing associations between groups of the same size (i.e., the same utility for data recipients).

The two observations above need to be considered if the data owner is interested in releasing more than one loose association among arbitrary subsets of fragments in $\mathcal{F}$. On the contrary, if the loose associations of interest operate on disjoint subsets of fragments (i.e., no fragment is involved in more than one loose association), they can be defined independently from each other without risks of unintended disclosure of sensitive associations.

Observation 3 *Given a fragmentation $\mathcal{F}$, and a set $\{\mathtt{F}_1,\ldots,\mathtt{F}_n\}$ of subsets of fragments in $\mathcal{F}$ (i.e.., $\mathtt{F}_i\subseteq\mathcal{F}$, $i=1,\ldots,n$), the release of n loose associations A_i, $i=1,\ldots,n$ is* safe *if $\forall i,j=1,\ldots,n$ with $i\neq j$, $\mathtt{F}_i\cap\mathtt{F}_j=\emptyset$.*

7 Related Work

Several research efforts have addressed the problem of protecting privacy in data publishing, proposing approaches based on either sanitizing (e.g., [6, 10, 11, 13–16, 20]) or fragmenting data (e.g., [1, 2, 4, 5, 7]) before their release. Our approach provides the same privacy guarantees as the well-known k-anonymity and ℓ-diversity (with ℓ=k) approaches, while releasing complete and truthful information thanks to the adoption of a different protection technique. Most fragmentation works, although showing similarities with our proposal, address a problem orthogonal to ours, as they aim at breaking sensitive associations while maximizing the ability of recipients of evaluating queries on fragments.

The works closest to ours complement fragmented data with information on their association, without disclosing sensitive information [7, 9, 21]. Our proposal is however more general, since these solutions operate on two fragments only, while we consider an arbitrary number of fragments when defining loose associations. Anatomy [21] considers the specific problem of protecting the association between respondents' identities and a sensitive attribute while our solution permits to protect any association among attributes. Also, Anatomy partitions the

original relation in groups of ℓ tuples before splitting attributes in two disjoint fragments. Hence, it can be considered a specific instance of our approach where k_l=1 and k_r=ℓ (or viceversa).

The work in [7] does not take into consideration the possible existence of duplicate values for non-key attributes, exposing therefore to frequency-based attacks sensitive associations on non-key attributes. Our heterogeneity properties overcome this issue, by preventing the presence of duplicates in the same group of tuples.

Our work may bring some resemblance with the proposals in [3, 18, 19]. The work in [19] adopts horizontal and vertical fragmentation to protect privacy of sparse multidimensional data (e.g., transactional data). The approach in [3] focuses instead on protecting recommendation data expressed by customers (i.e., Netflix movie ratings). Besides operating on different data models, both these proposals differ from our work since they are specifically targeted at protecting respondents' identities and their association with sensitive attributes. Also, they both adopt a dual approach with respect to loose associations, requiring homogeneity of values in fragments. The work in [18] addresses the problem of destroying the correlation between two disjoint subsets of attributes, preserving as much as possible the other correlations. Our approach does not aim at destroying correlations among attributes, as our goal is to preserve them as much as possible, while satisfying privacy constraints. Also, the solution in [18] adopts masking techniques, while our approach maintains data truthfulness.

Alternative approaches to protect privacy in data release are based on differential privacy [10, 11]. Although addressing a similar problem, differential privacy cannot be directly applied to the considered scenario. In fact, all these approaches introduce noise in the dataset that depends on the expected users queries. Our approach instead does not make assumptions on users queries and aims at releasing truthful data.

8 Conclusions

We presented an approach for extending the definition of loose association to multiple fragments. We first described the exposure risks that characterize the release of multiple loose associations between pairs of fragments, and then presented an approach supporting the definition of a loose association among an arbitrary number of fragments. We also discussed some properties of the proposed solution.

Acknowledgements. This work was supported in part by the EC within the 7FP under grant agreement 257129 (PoSecCo), by the Italian Ministry of Research within PRIN project "GenData 2020" (2010RTFWBH), and by Google, under the Google Research Award program.

References

1. Aggarwal, G., et al.: Two can keep a secret: A distributed architecture for secure database services. In: Proc. of CIDR 2005. Asilomar, CA, USA (January 2005)
2. Biskup, J.: Dynamic policy adaptation for inference control of queries to a propositional information system. JCS 20(5), 509–546 (2012)
3. Chang, C., Thompson, B., Wang, H., Yao, D.: Towards publishing recommendation data with predictive anonymization. In: Proc. of ASIACCS 2010. Beijing, China (April 2010)
4. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Combining fragmentation and encryption to protect privacy in data storage. ACM TISSEC 13(3), 22:1–22:33 (2010)
5. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Selective data outsourcing for enforcing privacy. JCS 19(3), 531–566 (2011)
6. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Samarati, P.: k-Anonymous data mining: A survey. In: Aggarwal, C., Yu, P. (eds.) Privacy-Preserving Data Mining: Models and Algorithms. Springer-Verlag (2008)
7. Cormode, G., Srivastava, D., Yu, T., Zhang, Q.: Anonymizing bipartite graph data using safe groupings. PVLDB 1(1), 833–844 (2008)
8. De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Livraga, G.: Enforcing subscription-based authorization policies in cloud scenarios. In: Proc. of DBSec 2012. Paris, France (July 2012)
9. De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Fragments and loose associations: Respecting privacy in data publishing. PVLDB 3(1), 1370–1381 (2010)
10. De Capitani di Vimercati, S., Foresti, S., Livraga, G., Samarati, P.: Protecting privacy in data release. In: Aldini, A., Gorrieri, R. (eds.) Foundations of Security Analysis and Design VI. Springer (2011)
11. Dwork, C.: Differential privacy. In: Proc. of ICALP 2006. Venice, Italy (July 2006)
12. Jhawar, R., Piuri, V., Samarati, P.: Supporting security requirements for resource management in cloud computing. In: Proc. of CSE 2012. Paphos, Cyprus (December 2012)
13. Kifer, D., Gehrke, J.: Injecting utility into anonymized datasets. In: Proc. of SIGMOD 2006. Chicago, IL, USA (June 2006)
14. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In: Proc. of ICDE 2007. Istanbul, Turkey (April 2007)
15. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: ℓ-Diversity: Privacy beyond k-anonymity. ACM TKDD 1(1), 3:1–3:52 (March 2007)
16. Raeder, T., Blanton, M., Chawla, N., Frikken, K.: Privacy-preserving network aggregation. In: Proc. of PAKDD 2010. Hyderabad, India (June 2010)
17. Samarati, P.: Protecting respondents' identities in microdata release. IEEE TKDE 13(6), 1010–1027 (2001)
18. Tao, Y., Pei, J., Li, J., Xiao, X., Yi, K., Xing, Z.: Correlation hiding by independence masking. In: Proc. of ICDE 2010. Long Beach, CA, USA (March 2010)
19. Terrovitis, M., Mamoulis, N., Liagouris, J., Skiadopoulos, S.: Privacy preservation by disassociation. PVLDB 5(10), 944–955 (2012)
20. Wang, K., Fung, B.: Anonymizing sequential releases. In: Proc. of KDD 2006. Philadelphia, PA, USA (August 2006)
21. Xiao, X., Tao, Y.: Anatomy: Simple and effective privacy preservation. In: Proc. of VLDB 2006. Seoul, Korea (September 2006)