Fragments and Loose Associations: Respecting Privacy in Data Publishing

Sabrina De Capitani di Vimercati
Università degli Studi di Milano
26013 Crema - Italy
sabrina.decapitani@unimi.it

Sara Foresti
Università degli Studi di Milano
26013 Crema - Italy
sara.foresti@unimi.it

Sushil Jajodia
George Mason University
Fairfax, VA 22030-4444
jajodia@gmu.edu

Stefano Paraboschi
Università di Bergamo
24044 Dalmine - Italy
parabosc@unibg.it

Pierangela Samarati
Università degli Studi di Milano
26013 Crema - Italy
pierangela.samarati@unimi.it

ABSTRACT

We propose a modeling of the problem of privacy-compliant data publishing that captures confidentiality constraints on one side and visibility requirements on the other side. Confidentiality constraints express the fact that some attributes, or associations among them, are sensitive and cannot be released. Visibility requirements express requests for views over data that should be provided. We propose a solution based on data fragmentation to split sensitive associations while ensuring visibility. In addition, we show how sensitive associations broken by fragmentation can be released in a sanitized form as *loose associations* formed in a way to guarantee a specified degree of privacy.

1. INTRODUCTION

In recent years, considerable attention has been devoted to the problem of guaranteeing privacy of sensitive data in databases that have to undergo public or semi-public release or be made available to third parties [5]. On one hand, today's society relies on the dissemination and sharing of information. On the other hand, there is a recognized and strong need to guarantee proper privacy protection of sensitive information.

Much research has focused on the data protection problem, investigating techniques providing different forms of protection and computing a "sanitized" version of the data for publication. Recently, most of this line of work has focused on k-anonymity and its variations (e.g., ℓ-diversity) for protecting respondents' identities and their sensitive information when releasing microdata (e.g., [4, 7, 8, 9]). Although some approaches have addressed the problem of safeguarding the utility of sanitized data, the problem of considering visibility requirements has not – to our knowledge – been investigated. Also, the modeling of the privacy problem, in the line of research mentioned above, typically assumes a setting where data to be protected are either quasi-identifiers or sensitive information associated with them, and provides protection by generalizing the values of quasi-identifying attributes. While important, k-anonymity and its variations capture only part of the problem.

In this paper, we propose a novel modeling of the problem of protecting privacy when publishing data that explicitly takes into consideration both privacy needs and visibility requirements. Our setting of the privacy problem is generic and does not assume, like typical k-anonymity solutions, a preliminary definition of identifying, quasi-identifying, and sensitive data. Rather, it supports the specification of *confidentiality constraints*, generically capturing privacy needs as sensitive attributes, or sensitive associations among them, that need to be protected. *Visibility requirements* provide an explicit means for data publishers and/or recipients to express the fact that certain data need to be published. Visibility requirements may come, for instance, from the need of third parties (e.g., research institutions) to which the data are released. Also, visibility requirements nicely permit capturing the fact that certain data are already available (e.g., from other external sources), avoiding publication of data whose combination with those already available might compromise privacy. Our solution is based on fragmenting data to break associations among them, guaranteeing respect of both confidentiality constraints and visibility requirements. We also put forward the idea of complementing fragments with *loose associations*. Intuitively, loose associations partition tuples within fragments in different groups and release association information at the group level, as opposed to releasing the actual tuple-to-tuple association. Our loose association problem is characterized by a privacy degree k defining the size of the association groups into which each actual association protected by a confidentiality constraint must be confused. We also define properties that the grouping of tuples has to satisfy to guarantee a given privacy degree k of the associations while maximizing the information released (i.e., minimizing the size of the association groups) and respecting all the given confidentiality constraints.

The remainder of the paper is organized as follows. Section 2 introduces the definition of confidentiality constraints and fragmentation. Section 3 formally defines visibility re-

Proceedings of the VLDB Endowment, Vol. 3, No. 1
Copyright 2010 VLDB Endowment 2150-8097/10/09... $ 10.00.

quirements. Section 4 introduces the definition of correct and minimal fragmentation. Section 5 introduces the concept of k-loose associations. Section 6 discusses how information exposure is influenced by the publication of k-loose associations. Section 7 discusses related work. Finally, Section 8 presents concluding remarks. The paper also includes three appendixes reporting respectively: an analysis of the fragment minimization problem and a SAT solver approach to its solution (Appendix A), experimental results showing the precision of the queries executed on k-loose associations (Appendix B), and the proofs of the theorems stated in the paper (Appendix C).

2. CONSTRAINTS AND SAFE FRAGMENTATION

We consider a scenario where the data to be protected are represented with a single relation s over a relation schema $S(a_1, \ldots, a_n)$. We use standard notations of relational database theory. Also, when clear from the context, we will use S to denote either the relation schema S or the set of attributes in S.

Privacy, or confidentiality, constraints express restrictions on the visibility, or on the joint visibility (association), of attributes in the relation. They are formally defined as follows [1, 2].

DEFINITION 2.1 (CONFIDENTIALITY CONSTRAINT). *Given a relation schema $S(a_1, \ldots, a_n)$, a* confidentiality constraint *c over S is a subset of the attributes in S.*

The semantics of confidentiality constraints is as follows. A singleton constraint states that an attribute is sensitive and therefore its values should not be visible; a non-singleton constraint states that an association among different attributes is sensitive and therefore the values of their attributes should not be visible in combination. Association constraints can reflect the sensitivity of the association itself or the fact that the association can cause inference of other sensitive information. Note how confidentiality constraints, while simple, capture different privacy requirements that may need to be expressed.

EXAMPLE 2.1. *Figures 1(a)-(b) illustrate relation* HOSPITAL *and the confidentiality constraints over it. Here, c_0 is a singleton constraint stating that the list of* `SSN`*s of patients is sensitive; c_1 and c_2 state that the associations between* `Patient` *and* `Illness`*, and between* `Patient` *and* `Doctor`*, respectively, are sensitive; c_3 and c_4 state that the associations among* `Birth`*,* `ZIP`*, and* `Illness`*, and among* `Birth`*,* `ZIP`*, and* `Doctor` *are sensitive* (*the rationale is that pair* ⟨`Birth`*,*`ZIP`⟩ *is a quasi-identifier [9] for patients and therefore constraints on the association of* `Patient` *with other attributes apply also to it*).

It is easy to see that the satisfaction of a confidentiality constraint c_i also implies the satisfaction of any other confidentiality constraint c_j such that $c_i \subseteq c_j$. Since constraints subsumed by others are redundant, we ignore them in the following and assume the set $\mathcal{C}$ of constraints to be *well defined*, that is, $\forall c_i, c_j \in \mathcal{C}$, $i \neq j$: $c_i \not\subseteq c_j$.

Confidentiality constraints can be enforced by properly fragmenting data, that is, by not including sensitive attributes in fragments and splitting sensitive associations among different fragments. We are then interested in fragmentations of the original relation. A fragmentation is a set of fragments, where each fragment is a subset of the attributes of the original relation. In other words, a fragment represents a (projection) view on the relation, and a fragmentation is a set of such views.

HOSPITAL

SSN	Patient	Birth	ZIP	Illness	Doctor
123-45-6789	Page	56/12/9	94142	hypertension	David
987-65-4321	Patrick	53/3/19	94141	gastritis	Daisy
246-81-3579	Patty	58/5/18	94139	flu	Damian
135-79-2468	Paul	53/12/9	94139	asthma	Daniel
975-31-8642	Pearl	56/12/9	94142	gastritis	Dorothy
864-29-7531	Philip	57/6/25	94141	obesity	Drew
246-89-7531	Phoebe	60/7/25	94142	measles	Dennis
135-79-8642	Piers	53/12/1	94140	hypertension	Daisy

(a)

$c_0 = \{\texttt{SSN}\}$
$c_1 = \{\texttt{Patient,Illness}\}$
$c_2 = \{\texttt{Patient,Doctor}\}$
$c_3 = \{\texttt{Birth,ZIP,Illness}\}$
$c_4 = \{\texttt{Birth,ZIP,Doctor}\}$

(b)

$v_1 = \texttt{Patient} \lor \texttt{ZIP}$
$v_2 = (\texttt{Birth} \land \texttt{ZIP}) \lor \texttt{SSN}$
$v_3 = \texttt{Illness} \land \texttt{Doctor}$

(c)

F_l

	Birth	ZIP
l_1	56/12/9	94142
l_2	53/3/19	94141
l_3	58/5/18	94139
l_4	53/12/9	94139
l_5	56/12/9	94142
l_6	57/6/25	94141
l_7	60/7/25	94142
l_8	53/12/1	94140

F_r

Illness	Doctor	
hypertension	David	r_1
gastritis	Daisy	r_2
flu	Damian	r_3
asthma	Daniel	r_4
gastritis	Dorothy	r_5
obesity	Drew	r_6
measles	Dennis	r_7
hypertension	Daisy	r_8

(d)

Figure 1: A plaintext relation (a), confidentiality constraints (b), visibility requirements (c), and fragmentation (d)

DEFINITION 2.2 (FRAGMENTATION). *Given a relation schema S, a* fragmentation *$\mathcal{F}$ of S is a set of fragments $\mathcal{F}=\{F_1, \ldots, F_m\}$ such that $\forall F \in \mathcal{F}$, $F \subseteq S$.*

Note that, according to our definition, a fragmentation does not need to be *complete*, that is, it does not need to include all the attributes of the original relation (our fragments are therefore different from fragments in [1, 2]). A fragment instance, denoted f, of a fragment F of S is the set of tuples of relation s over S projected on the attributes in F. We assume possible duplicates to be maintained in fragment instances, that is, fragment instances have the same cardinality as the original relation. The reason for this is to frame our problem in the most general setting which, from a protection point of view, exposes more information (the cardinality of occurrences of the values of attributes participating in a confidentiality constraint is exposed). For simplicity, in the following, when clear from the context, we refer to fragment instances simply using the term fragments.

A fragmentation is *safe* if the information it releases does not violate the constraints. In other words, a safe fragmentation should not allow visibility of sensitive attributes or sensitive associations, neither directly (in a single fragment) nor indirectly (by joining fragments). This is formally stated by the following definition.

Definition 2.3 (Safe fragmentation). *Given a fragmentation $\mathcal{F}$ over a relation schema S and a set $\mathcal{C}$ of confidentiality constraints over S, we say that $\mathcal{F}$ is* safe *with respect to $\mathcal{C}$ iff both the following conditions hold:*

1. *$\forall F\in\mathcal{F}$, $\forall c\in\mathcal{C}$: $c\not\subseteq F$;*
2. *$\forall F_i,F_j\in\mathcal{F}$, $i\neq j$: $F_i\cap F_j=\emptyset$.*

Condition 1 ensures direct obedience of all the confidentiality constraints (no explicit visibility of sensitive attributes or associations), and condition 2 ensures indirect obedience (the absence of attributes in common between fragments prevents joins on fragments to retrieve associations). Figure 1(d) illustrates fragmentation $\mathcal{F}$={{Birth,ZIP}, {Illness,Doctor}} over the relation in Figure 1(a), which is safe with respect to the constraints in Figure 1(b). Note that, for the sake of readability, the semi-tuples ($l_1,\ldots,l_8$ and $r_1,\ldots,r_8$) in the two fragments have been reported following the same order of the tuples in the original relation hospital.

3. VISIBILITY REQUIREMENTS

Visibility requirements express views over data that the fragmentation should satisfy. Views can express that certain attributes should be visible or that certain attributes should be released in conjunction, meaning that their association, not only their individual values, should be released. Views can also specify alternative visibility options over the data, giving different choices on the attributes, or sets of attributes, that can be released (provided that at least one of the options is satisfied).

With a very general setting, we assume that a visibility requirement can be any monotonic boolean formula over attributes of the original relation schema.

Definition 3.1 (Visibility requirement). *Given a relation schema $S(a_1,\ldots,a_n)$, a* visibility requirement *v over S is a monotonic boolean formula over $\{a_1,\ldots,a_n\}$.*

The reason for considering only monotonic formulas is that negations over attributes correspond to requests for non-visibility over some attributes, and are therefore captured by confidentiality constraints. Note that, besides ensuring clarity of the specifications, the clear separation between visibility requirements and confidentiality constraints is a desirable design feature. In fact, in many real-world scenarios, the specification of confidentiality constraints on one side, and the specification of desired views of data to be published on the other side, belong to different authorities.

Intuitively, visibility requirements impose the inclusion, or joint inclusion, of attributes in fragments of a fragmentation. The semantics of a visibility requirement is therefore easily explained with reference to fragments. Let v be a visibility requirement and $\mathcal{F}$ be a fragmentation:

- $v=a$ is satisfied if attribute a belongs to a fragment (i.e., $\exists F\in\mathcal{F}$: $a\in F$);
- $v=v_i\wedge v_j$ is satisfied iff v_i and v_j are satisfied by the *same* fragment (e.g., $v=a_1\wedge a_2$ is satisfied iff $\exists F\in\mathcal{F}$: $a_1,a_2\in F$);
- $v=v_i\vee v_j$ is satisfied if at least one of v_i or v_j is satisfied by a fragment (e.g., $v=a_1\vee a_2$ is satisfied iff $\exists F\in\mathcal{F}$: $a_1\in F$ or $a_2\in F$).

Example 3.1. *Figure 1(c) reports possible visibility requirements over relation* Hospital *in Figure 1(a). Here, v_1 states that either* Name*s of patients or their* ZIP *codes should be released; v_2 states that either* Birth *dates and* ZIP *codes of patients in association should be released or the* SSN *of patients should be released; v_3 states that* Illness*es and* Doctor*s, as well as their association, should be released.*

The semantics of a set of visibility requirements is that *all* the requirements should be satisfied, not necessarily by a single fragment. Note the difference between stating two visibility requirements v_i,v_j as: *1)* two separate requirements v_i,v_j, meaning that *both* v_i and v_j should be satisfied by the same or by a different fragment; *2)* a single (and-ed) requirement $v_z=v_i\wedge v_j$, meaning that *both* v_i and v_j should be satisfied by the *same* fragment; and *3)* a single (or-ed) requirement $v_z=v_i\vee v_j$, meaning that *at least one* of v_i or v_j should be satisfied by a fragment.

By interpreting a fragment F as a conjunction over the attributes composing the fragment (interpreting attributes as boolean variables), satisfiability of a visibility requirement can be expressed in terms of the usual implication (→) between logic formulas and can be formally defined as follows.

Definition 3.2 (Satisfies). *Given a relation schema $S(a_1,\ldots,a_n)$ and a set $\mathcal{V}$ of visibility requirements over S, a fragmentation $\mathcal{F}$ of S* satisfies *$\mathcal{V}$, denoted $\mathcal{F}\to\mathcal{V}$, iff $\forall v\in\mathcal{V}$, $\exists F\in\mathcal{F}$: $F\to v$.*

The fragmentation in Figure 1(d) satisfies the visibility requirements in Figure 1(c), since $F_l\to v_1$, $F_l\to v_2$, and $F_r\to v_3$.

4. CORRECT AND MINIMAL FRAGMENTATION

Given a relation, a set of confidentiality constraints, and a set of visibility requirements, our problem is to determine a *correct* fragmentation, that is, a fragmentation that is safe with respect to the constraints and satisfies the visibility requirements. Correctness is formally defined as follows.

Definition 4.1 (Correctness). *Given a relation schema $S(a_1,\ldots,a_n)$, a set $\mathcal{C}$ of confidentiality constraints over S, and a set $\mathcal{V}$ of visibility requirements over S, a fragmentation $\mathcal{F}$ of S is* correct *wrt $\mathcal{C}$ and $\mathcal{V}$ iff: $\mathcal{F}$ is safe with respect to $\mathcal{C}$ (Definition 2.3), and $\mathcal{F}$ satisfies $\mathcal{V}$ (Definition 3.2).*

Also, we aim at a *minimal* fragmentation, that is, a fragmentation that minimizes the number of fragments. Indeed, avoiding splitting attributes when not needed for satisfying the constraints is convenient, as it maximizes the visibility over the data. In fact, maintaining attributes together in a fragment releases not only their values but their association, which, if not protected (directly or indirectly) by confidentiality constraints, can be safely released. Our problem is then formally defined as follows.

Problem 4.1 (Min-CF). *Given a relation schema $S(a_1,\ldots,a_n)$, a set $\mathcal{C}$ of confidentiality constraints over S, and a set $\mathcal{V}$ of visibility requirements over S, determine (if it exists) a fragmentation $\mathcal{F}$ such that:*

1. *$\mathcal{F}$ is a* correct fragmentation *(Definition 4.1);*

2. $\nexists\mathcal{F}'$ s.t. $\mathcal{F}'$ is correct and the number of fragments of $\mathcal{F}'$ is less than the number of fragments of $\mathcal{F}$.

Theorem 4.1. *The* Min-CF *problem is NP-hard.*

Our approach to solve the Min-CF problem, identifying the optimum solution, is based on the public availability of SAT solvers (see Appendix A), which support the efficient resolution of SAT problems even with millions of variables.

Since we are clearly interested only in fragmentations that are *correct and minimal*, in the following we simply use the term fragmentation to refer to a fragmentation that satisfies both properties.

5. PUBLISHING LOOSE ASSOCIATIONS

While fragments, by definition, cannot be joined (as this would imply a violation of confidentiality constraints), in this section we put forward the idea of publishing a loose association among their tuples (sub-tuples of the original relation) to release some information on the association existing in the original relation, provided a given *privacy degree* of the association is respected. Intuitively, our loose associations hide tuples participating in the associations in groups and provide information on the associations only at the group level. Loose associations, while not impacting privacy (as dictated by the privacy degree) provide enriched utility of the published data, supporting, for example, aggregate queries and data mining. In the following, we focus on the problem of publishing a loose association between a pair of fragments, denoted F_l and F_r (left and right fragment, respectively), in $\mathcal{F}$.

In this paper, we assume the absence of "external knowledge", that is, of additional information describing the relationship between values in the two fragments and that could be exploited for the reconstruction of some of the original associations. We explicit this requirement by assuming that attributes in the two fragments are independent (i.e., their relationship is not known by the adversary, in a statistical sense, apart from what can be known by the observation of the published data).

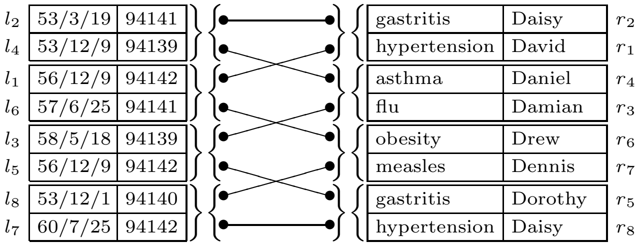
5.1 k-grouping

We start by defining grouping over fragment instances. Since the size of the groups into which tuples in fragments are clustered impacts our problem, we characterize a grouping with an index denoting the lower bound on the size that groups may have.

Definition 5.1 (k-Grouping). *Given a fragment instance f_i and a set GID_i of group identifiers, a* k-grouping *over f_i is a surjective function $\mathcal{G}_i$:$f_i \rightarrow \mathsf{GID}_i$ such that $\forall g_i \in \mathsf{GID}_i :\mid \mathcal{G}_i^{-1}(g_i) \mid \geq k$.*

A k-grouping function associates with each tuple in a fragment a group identifier in such a way that each group has at least k tuples mapping to it. A k-grouping is *minimal* if it minimizes the size of the groups, provided that k is respected, or, equivalently, it maximizes the number of groups into which tuples are mapped. Formally, a k-grouping over a fragment f_i is minimal iff the cardinality of the image of $\mathcal{G}_i$ over f_i is equal to the floor of the ratio between the cardinality of f_i and k, that is, $|\mathsf{GID}_i| = \left\lfloor \frac{|f_i|}{k} \right\rfloor$.

In the following, we consider the problem of grouping two fragments f_l and f_r and we will refer to a (k_l,k_r)-grouping to denote with a single term the two components: a k_l-grouping over f_l and a k_r-grouping over f_r. A (k_l,k_r)-grouping is said to be *minimal* iff both its grouping components are minimal. Figure 2 illustrates an example of minimal (2,2)-grouping over the fragments in Figure 1(d), where each group contains two tuples. For simplicity, given a tuple t_i in relation s over S, we denote with l_i the sub-tuple $t_i[F_l]$ in fragment f_l and with r_i the sub-tuple $t_i[F_r]$ in fragment f_r. The grouping of two fragments, together with the original relation specifying associations among the tuples in fragments, induces an association among groups, formally defined as follows.



l_2	53/3/19	94141	gastritis	Daisy	r_2
l_4	53/12/9	94139	hypertension	David	r_1
l_1	56/12/9	94142	asthma	Daniel	r_4
l_6	57/6/25	94141	flu	Damian	r_3
l_3	58/5/18	94139	obesity	Drew	r_6
l_5	56/12/9	94142	measles	Dennis	r_7
l_8	53/12/1	94140	gastritis	Dorothy	r_5
l_7	60/7/25	94142	hypertension	Daisy	r_8

Figure 2: An example of (2,2)-grouping

Definition 5.2 (Group association). *Given a fragmentation $\mathcal{F}$ of S, a relation s over S, and two grouping functions $\mathcal{G}_l$, $\mathcal{G}_r$ defined for f_l and f_r over F_l, F_r in $\mathcal{F}$, a* group association *$A \subseteq \mathsf{GID}_l \times \mathsf{GID}_r$ over f_l and f_r is a set of pairs such that:*

- *$|s|$=$|A|$;*
- *it is possible to define a bijective mapping between s and A that associates each $t \in s$ with a pair ($\mathcal{G}_l(l)$,$\mathcal{G}_r(r)$)$\in A$.*

Figure 2 graphically illustrates the group association induced by the (2,2)-grouping by means of edges connecting black dots that correspond to tuples in the groups. Intuitively, each edge represents the association between a tuple in a group of the left fragment with a tuple in a group of the right fragment.

Note that Definition 5.2 assumes a one-to-one correspondence between tuples in the original relation and corresponding semi-tuples in each fragment. This is consistent with the fact that possible duplicates in fragments are maintained and fragments therefore have the same cardinality as the original relation.

The protection offered by a group association can be compromised by the presence, within a group, of tuples that have the same values over attributes whose association with some attributes in the other fragment is sensitive. It is not sufficient to guarantee that groups do not contain duplicate tuples. Indeed, there can be tuples that, although different, have the same values for the attributes involved in a confidentiality constraint. The following definition captures the relationship among tuples carrying the same values for attributes that, together with the other fragment, would cover a confidentiality constraint.

Definition 5.3 (Alike). *Given a set $\mathcal{C}$ of confidentiality constraints, two fragments F_l and F_r, and their instances f_l and f_r, two tuples l_i,$l_j$$\in$$f_l$ (r_i,$r_j$$\in$$f_r$, resp.)*

F_l

	Birth	ZIP	G
l_1	56/12/9	94142	bz2
l_2	53/3/19	94141	bz1
l_3	58/5/18	94139	bz3
l_4	53/12/9	94139	bz1
l_5	56/12/9	94142	bz3
l_6	57/6/25	94141	bz2
l_7	60/7/25	94142	bz4
l_8	53/12/1	94140	bz4

(a)

A

G$_l$	G$_r$
bz1	id1
bz1	id2
bz2	id1
bz2	id3
bz3	id2
bz3	id4
bz4	id3
bz4	id4

(b)

F_r

Illness	Doctor	G	
hypertension	David	id1	r_1
gastritis	Daisy	id1	r_2
flu	Damian	id2	r_3
asthma	Daniel	id2	r_4
gastritis	Dorothy	id4	r_5
obesity	Drew	id3	r_6
measles	Dennis	id3	r_7
hypertension	Daisy	id4	r_8

(c)

Figure 3: An example of 4-loose association

are said to be alike *wrt a constraint $c \in \mathcal{C}$, with $c \subseteq F_l \cup F_r$, denoted $l_i \simeq_c l_j$ ($r_i \simeq_c r_j$, resp.), iff $l_i[c \cap F_l] = l_j[c \cap F_l]$ ($r_i[c \cap F_r] = r_j[c \cap F_r]$, resp.). Two tuples are said to be alike wrt a set $\mathcal{C}$ of constraints, denoted $t_i \simeq_{\mathcal{C}} t_j$, if they are alike wrt at least one constraint $c \in \mathcal{C}$.*

According to Definition 5.3, two tuples in fragment f_l (f_r, resp.) are alike wrt a constraint c covered by the two fragments, if they have the same values for the attributes of F_l (F_r, resp.) appearing in c. For instance, with respect to the fragments in Figure 2, $l_1 \simeq_{c_3} l_5$. Note that the reason for considering only confidentiality constraints completely covered by the two fragments is that, by definition, all the other confidentiality constraints cannot be violated by merging the two fragments (as at least one attribute would be missing). Since the set $\mathcal{C}$ of confidentiality constraints is given, in the following we omit $\mathcal{C}$ as a subscript of the alike relationship between tuples (i.e., we write $t_i \simeq t_j$ instead of $t_i \simeq_{\mathcal{C}} t_j$).

5.2 k-loose associations

We are now ready to define our concept of loose association, characterized by a degree k of protection, over an association induced by a grouping.

Definition 5.4 (k-Looseness). *Given a set $\mathcal{C}$ of confidentiality constraints and a group association A over f_l and f_r, A is said to be* k-loose *iff:*

- $\forall g_l \in \mathsf{GID}_l : T = \bigcup_z \{\mathcal{G}_r^{-1}(g_z) \mid (g_l, g_z) \in A\} \implies |T| \geq k$, and $\forall r_i, r_j \in T, i \neq j : r_i \not\simeq r_j$;
- $\forall g_r \in \mathsf{GID}_r : T = \bigcup_z \{\mathcal{G}_l^{-1}(g_z) \mid (g_z, g_r) \in A\} \implies |T| \geq k$, and $\forall l_i, l_j \in T, i \neq j : l_i \not\simeq l_j$.

According to Definition 5.4, an association is k-loose iff for each group g_l in the left (group g_r in the right, resp.) fragment, the union of the tuples in all groups g_z with which g_l (g_r, resp.) is associated is a set that has cardinality at least k and that does not contain any two tuples that are alike. Intuitively, an association is k-loose iff for each real association existing in the original relation it releases at least k possible distinct associations.

Figure 3 illustrates the 4-loose association induced by the (2,2)-grouping in Figure 2. The k-loose association is published as a relation A(G$_l$,G$_r$) whose tuples correspond to pairs (g_{l_i},g_{r_j}) in A. Also, fragments are enriched with an attribute G, reporting, for each tuple $l \in f_l$ ($r \in f_r$, resp.), the group to which the tuple belongs.

Clearly, a k-loose association is also k'-loose for any $k' \leq k$. For lower privacy requirements (i.e., smaller k), however, a more precise information on the associations, that is, working on smaller groups, would suffice. Since the main reason for publishing loose associations is to provide information on the original relation, provided that a degree of protection k is guaranteed, smaller groups, which imply more precise information, should then be preferred. Our goal is therefore to determine a *minimal k-loose association*. Since the association is induced by the groupings over the two involved fragments, minimality of the association means requiring minimality of the corresponding groupings, as formally stated by the following problem.

Problem 5.1 (Min k-loose). *Given a set $\mathcal{C}$ of confidentiality constraints, a fragmentation $\mathcal{F}$ of S, a relation s over S, two fragments F_l, F_r in $\mathcal{F}$, their instances f_l and f_r, and a privacy degree k, determine a* minimal (k_l,k_r)-grouping *such that:*

- *the induced group association A is k-loose;*
- *$\nexists$ a (k'_l,k'_r)-grouping over f_l and f_r, with $k'_l \cdot k'_r < k_l \cdot k_r$, such that the induced group association A' is k-loose.*

Theorem 5.1. *The* Min k-loose *problem is NP-hard.*

Note that the Min k-loose problem may not always have a solution. This happens, for example, when there are sensitive values appearing with more than $\frac{|s|}{k}$ occurrences. If few sensitive values cause the problem, the corresponding tuples can be suppressed, like in k-anonymity approaches [3, 7, 9].

5.3 (k_l, k_r)-grouping and k-looseness

As it is clear from Definition 5.4, there is a correspondence between the degree of the groupings and the degree of k-looseness that the induced group association can provide. Trivially, a (k_l,k_r)-grouping cannot certainly provide k-looseness for a $k > k_l \cdot k_r$. Whether it provides k-looseness for lower values of k depends on how the groups are defined. In the following, we introduce three properties of grouping whose satisfaction ensures satisfaction of k-looseness with a minimal grouping.

The first property we introduce is heterogeneity within each group.

Property 5.1 (Group heterogeneity). *Given a set $\mathcal{C}$ of confidentiality constraints, two fragments F_l and F_r in $\mathcal{F}$, and their instances f_l and f_r, grouping functions $\mathcal{G}_l$ over f_l and $\mathcal{G}_r$ over f_r satisfy* group heterogeneity *iff $\forall f_i \in \{f_l, f_r\}$, $\forall t_z, t_w \in f_i$: $t_z \simeq t_w \implies \mathcal{G}_i(t_z) \neq \mathcal{G}_i(t_w)$.*

Group heterogeneity ensures diversity of the tuples appearing in the groups with respect to the attributes involved in the confidentiality constraints covered by the fragments. For instance, the (2,2)-grouping in Figure 2 satisfies Property 5.1, since all the groups of the left fragment, as well as all the groups of the right fragment, have different values for the attributes appearing in constraints c_3 and c_4 in Figure 1(b). The cardinality of a heterogeneous group provides a measure of diversity of the group, that is, of the number of different values for attributes participating in a confidentiality constraint.

The second property we introduce is heterogeneity of the groups with which each group is associated (in the induced association).

Property 5.2 (Association heterogeneity). *A group association A satisfies* association heterogeneity *iff* $\forall(g_i,g_z),(g_j,g_w)\in A$:

- $i=j \Longrightarrow z \neq w$;
- $z=w \Longrightarrow i \neq j$.

Intuitively, association heterogeneity guarantees that the group association does not contain duplicates. Association heterogeneity of a (k_l,k_r)-grouping ensures that for each real tuple in the original relation there are at least $k_l{\cdot}k_r$ pairs in the group association that may correspond to it. The group association in Figure 2 satisfies Property 5.2, since there is at most one edge between each pair of groups.

Association heterogeneity provides only a superficial inequality among associations. As a matter of fact, if the different groups associated with a given group have tuples which are alike (Definition 5.3), the $k_l{\cdot}k_r$ associated tuples do not correspond to $k_l{\cdot}k_r$ different values for the attributes involved in a constraint covered by the fragments.

Our third property captures the need for heterogeneity of the groups associated with the same group.

Property 5.3 (Deep heterogeneity). *Given a set $\mathcal{C}$ of confidentiality constraints, a group association A over f_l and f_r satisfies* deep heterogeneity *iff* $\forall(g_i,g_z),(g_j,g_w)\in A$:

- $i=j \Longrightarrow \nexists r_z,r_w$: $r_z\in\mathcal{G}_r^{-1}(g_z),r_w\in\mathcal{G}_r^{-1}(g_w)$, $r_z\simeq r_w$;
- $z=w \Longrightarrow \nexists l_i,l_j$: $l_i\in\mathcal{G}_l^{-1}(g_i),l_j\in\mathcal{G}_l^{-1}(g_j)$, $l_i\simeq l_j$.

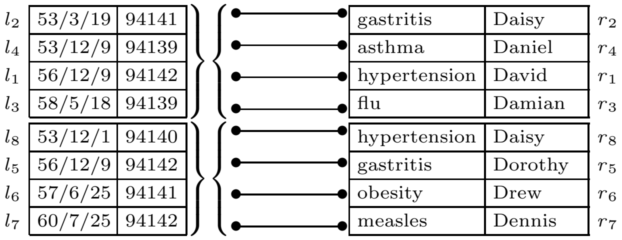
Deep heterogeneity requires that the association induced by the grouping be such that no group is associated with two groups that contain alike tuples. The (2,2)-grouping in Figure 2 satisfies Property 5.3, since each group of the left (right, resp.) fragment is associated with two groups of the right (left, resp.) fragment that contain tuples with different values for attributes Illness and Doctor (for the pair ⟨Birth,ZIP⟩, resp.).

If a (k_l,k_r)-grouping and its induced association satisfy the three properties above, the induced association is guaranteed to be k-loose with $k\leq k_l{\cdot}k_r$. If the (k_l,k_r)-grouping is minimal, the association is a minimal k-loose association for $k{=}k_l{\cdot}k_r$. This is captured by the following theorem.

Theorem 5.2. *Given a set $\mathcal{C}$ of confidentiality constraints, a fragmentation $\mathcal{F}$ of S, a relation s over S, two fragments F_l and F_r in $\mathcal{F}$, their instances f_l and f_r, and a minimal (k_l,k_r)-grouping that satisfies Properties 5.1, 5.2, and 5.3, then the group association A induced by the (k_l,k_r)-grouping is k-loose (Definition 5.4) for each $k\leq k_l{\cdot}k_r$, and is a minimal k-loose for $k{=}k_l{\cdot}k_r$.*

Theorem 5.2 provides us with the nice property that, to satisfy a given degree of k-looseness, any (k_l,k_r)-grouping satisfying the three properties above and such that $k\leq k_l{\cdot}k_r$ would work. For instance, k-looseness of 12 could be provided, among other choices, with a (4,3)-grouping or with a (6,2)-grouping; clearly even a (12,1)-grouping would work.

The case where either k_l or k_r is equal to 1 deserves a separate mention. We make the note for the case where $k_r{=}1$ (the case for $k_l{=}1$ is analogous). Since in a $(k,1)$-grouping the right fragment is split into singleton groups, by definition, the group association is such that no two groups of the left fragment are associated with the same group of the right fragment. Given this property, we refer to a grouping where either k_l or k_r is equal to 1 as a *flat grouping.* Figure 4 illustrates an example of flat grouping with $k_l{=}4$. Intuitively, a flat grouping corresponds to slicing the original relation into sets of tuples of size at least k and publishing, for what concerns the associations, instead of the exact values for the attributes in the left fragment, the groups into which they are mapped (exact values do remain available in the fragments). Note that since our approach protects the association, the k-looseness applies to the association as a whole (i.e., also to the right fragment). In fact, in the association induced by a $(k,1)$-grouping respecting the three properties above, each tuple in the right fragment will also be associated with at least k tuples in the left fragment. It is nice to see how a $(k,1)$-grouping resembles the approach of k-anonymity [9] (slicing the original relation in different clusters and generalizing part of it) but, working on associations and thanks to the heterogeneity properties it also captures at the same time the concept of ℓ-diversity [8] (with $\ell{=}k$). Apart from the analogy for this particular case, we note that our problem and solution are fundamentally different from the k-anonymity problem: in our approach, the values appearing in the original relation are published at the detailed level in the fragments, while it is the association that is obfuscated.



l_2	53/3/19	94141	gastritis	Daisy	r_2
l_4	53/12/9	94139	asthma	Daniel	r_4
l_1	56/12/9	94142	hypertension	David	r_1
l_3	58/5/18	94139	flu	Damian	r_3
l_8	53/12/1	94140	hypertension	Daisy	r_8
l_5	56/12/9	94142	gastritis	Dorothy	r_5
l_6	57/6/25	94141	obesity	Drew	r_6
l_7	60/7/25	94142	measles	Dennis	r_7

Figure 4: An example of (4,1)-grouping

When neither k_l nor k_r is equal to one, our loose associations result *sparse* (see Figure 2). Maintaining the size of groups small, sparse grouping guarantees larger applicability, while granting the same level of protection as flat grouping. Intuitively, being the alike relationship non-transitive when at least two constraints are involved, a sparse grouping might find a solution even when a flat grouping does not. When only one constraint is involved, and therefore the alike relationship is transitive, sparse grouping and flat grouping are equally applicable (i.e., if there is a solution for one there is also a solution for the other and viceversa). The following theorem formalizes this relationship among the flat and sparse groupings.

Theorem 5.3. *Given two fragments F_l and F_r in $\mathcal{F}$, their instances f_l and f_r, a privacy degree k, and a number n of constraints c such that $c\subseteq F_l\cup F_r$, then*

1. *if $n=1$: $\exists$ a flat grouping providing k-looseness $\Longleftrightarrow$ $\exists$ a sparse grouping providing k-looseness;*
2. *if $n>1$:*
   - *(a) $\exists$ a flat grouping providing k-looseness $\Longrightarrow$ $\exists$ a sparse grouping providing k-looseness;*
   - *(b) $\exists$ a sparse grouping providing k-looseness $\not\Longleftarrow$ $\exists$ a flat grouping providing k-looseness.*

6. DISCUSSION

The publication of loose associations, in addition to fragments, increases the *utility* of data publication, at the same time clearly bringing some *exposure* of sensitive associations and therefore decreasing *privacy*. As a matter of fact, publishing loose associations provides some information on the possible combinations of tuples in the different fragments, as it restricts the possible combinations to those allowed by the loose associations. In this section, we discuss the probability that an observer can establish on the fact that an association between values exists in the original relation, depending on whether she has knowledge of the fragments only or also of the loose associations.

We first introduce some notations and describe how to model the exposure of sensitive associations in terms of probabilities. For each α=(l_i,r_j) expressing the association of tuple l_i in f_l with a tuple r_j in f_r, we denote with $\mathcal{P}(\alpha \in s|f_l,f_r)$ the probability of association α to be present in the original relation s, given the knowledge of fragments f_l and f_r. We denote with $\mathcal{P}(\alpha \in s|f_l,f_r,A)$ the probability of association α to be present in the original relation s, given the knowledge of fragments f_l and f_r and of the loose association A. Since our discussion focuses on given s, f_l, and f_r, to simplify notation, in the following we will write $\mathcal{P}(\alpha)$ as a shorthand for $\mathcal{P}(\alpha \in s|f_l,f_r)$ and $\mathcal{P}^A(\alpha)$ as a shorthand for $\mathcal{P}(\alpha \in s|f_l,f_r,A)$.

Our goal is to protect sensitive associations as defined in the confidentiality constraints and therefore we need to worry about the probability of the associations among tuples and especially of their sensitive values. For instance, if two tuples l_i, l_j in the left fragment are alike wrt a given constraint c, the probability of exposing an association of their sensitive values wrt c is, for all r in f_r, the composition of the probability that: *1)* l_i is associated with tuple r, and *2)* l_j is associated with tuple r. By exploiting the independence assumption, such a probability is $\mathcal{P}(l_i,r)$ + $\mathcal{P}(l_j,r)$ − ($\mathcal{P}(l_i,r) \cdot \mathcal{P}(l_j,r)$). We then model the probability of exposing values that are sensitive wrt a given constraint c as a probability among sets of tuples that are alike wrt c. Analogous reasoning applies to $\mathcal{P}^A()$, that is, to the exposure of sensitive values of alike tuples if a loose association A is published. Given a constraint c, we extend the notion of probabilities over equivalence classes of tuples that are alike wrt the constraint. In the following, we denote with $\mathcal{P}(L,R)$ and $\mathcal{P}^A(L,R)$ the composite probability (for $\mathcal{P}()$ and $\mathcal{P}^A()$) of the independent event (l,r) for each $l \in L$, $r \in R$, where L and R are equivalence classes of tuples that are alike, in their corresponding fragment, wrt c.

We can now evaluate the exposure of associations among tuples and values without or with a k-loose association. If no association is published, but only the fragments are, every tuple in a fragment is equally likely to be associated with any other tuple in the other fragment, that is, $\mathcal{P}(l_i,r_j)$= $1/|s|$, for each $l_i \in f_l$, $r_j \in f_r$. For instance, with respect to the fragments in Figure 1(d), $\mathcal{P}(l_1,r_1)$ = 1/8. The probability of associating values appearing in a fragment with values appearing in the other fragment (and together covering a constraint c), depends on whether the fragments contain tuples that are alike wrt c. Basically, tuples that are alike wrt c have to be considered together, since the exposure of the sensitive values in them is the composite probability over the set of such tuples, as discussed above. Figure 5 illustrates the exposure of the association, that is, the probability of the association actually holding in the original relation for tuples (Figure 5(a)) and for sensitive values wrt c_3 (Figure 5(b)), respectively. Each row (column, resp.) corresponds to a tuple (Figure 5(a)) or to a set of tuples alike wrt c_3 (Figure 5(b)) in the left (right, resp.) fragment. Each entry reports the probability that the association between the tuple (or set of tuples) in the row with the tuple (or set of tuples) in the column exists in the original relation.

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
l_1	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_2	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_3	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_4	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_5	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_6	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_7	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_8	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$

(a)

	r_1,r_8	r_2,r_5	r_3	r_4	r_6	r_7
l_1,l_5	$\frac{1695}{4096}$	$\frac{1695}{4096}$	$\frac{15}{64}$	$\frac{15}{64}$	$\frac{15}{64}$	$\frac{15}{64}$
l_2	$\frac{15}{64}$	$\frac{15}{64}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_3	$\frac{15}{64}$	$\frac{15}{64}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_4	$\frac{15}{64}$	$\frac{15}{64}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_6	$\frac{15}{64}$	$\frac{15}{64}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_7	$\frac{15}{64}$	$\frac{15}{64}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
l_8	$\frac{15}{64}$	$\frac{15}{64}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$

(b)

Figure 5: Probabilities of associations between tuples (a) and values alike wrt c_3 (b) if no association is published

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
l_1	$\frac{1}{4}$	$\frac{1}{4}$	-	-	-	$\frac{1}{4}$	-	$\frac{1}{4}$
l_2	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-	-	-	-
l_3	-	-	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-	$\frac{1}{4}$	-
l_4	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-	-	-	-
l_5	-	-	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-	$\frac{1}{4}$	-
l_6	$\frac{1}{4}$	$\frac{1}{4}$	-	-	-	$\frac{1}{4}$	-	$\frac{1}{4}$
l_7	-	-	-	-	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
l_8	-	-	-	-	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

(a)

	r_1,r_8	r_2,r_5	r_3	r_4	r_6	r_7
l_1,l_5	$\frac{7}{16}$	$\frac{7}{16}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
l_2	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-	-
l_3	-	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-	$\frac{1}{4}$
l_4	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-	-
l_6	$\frac{7}{16}$	$\frac{1}{4}$	-	-	$\frac{1}{4}$	-
l_7	$\frac{1}{4}$	$\frac{1}{4}$	-	-	$\frac{1}{4}$	$\frac{1}{4}$
l_8	$\frac{1}{4}$	$\frac{1}{4}$	-	-	$\frac{1}{4}$	$\frac{1}{4}$

(b)

Figure 6: Probabilities of associations between tuples (a) and values alike wrt c_3 (b) if the 4-loose association of Figure 3 is published

The publication of a k-loose association reduces the uncertainty over the associations actually belonging to the original relation, as it allows discarding associations not possible according to the released k-loose association. The probability of associating a tuple in a fragment with a tuple in another fragment, given the publication of a k-loose association, is at most $1/k$. For instance, Figure 6 illustrates the probability of the association actually holding in the original relation for tuples (Figure 6(a)) and for sensitive values wrt constraint c_3 (Figure 6(b)), respectively, when the 4-loose association in Figure 3 is released. Symbol '-' is used for cells whose values are equal to 0. Note that, in the case of sparse grouping, the distribution of the probabilities of associating tuples in the two fragments results sparse. However, each row and each column of the table reporting such probabilities (e.g., Figure 6(a)) has k occurrences of a $1/k$ probability.

The *utility* of publishing a loose association can then be estimated by computing the average over the variation of probability, that is, $|\mathcal{P}^A(L_i,R_j) - \mathcal{P}(L_i,R_j)|$ for each association. In this way, a threshold $\delta_{\max}$ regulating the maximum increase of exposure allowed can be specified and k-loose association A could be safely published only if $\delta_{\max} \geq (\mathcal{P}^A(L_i,R_j) - \mathcal{P}(L_i,R_j))$, for all tuples, or sets of alike tuples L_i, R_j in the left and right fragments, respectively.

We evaluated the utility of loose associations in terms of the precision in responding to queries. We performed several experiments with a varying degree of k looseness and different query workloads (see Appendix B). As expected, the experiments show that the level of precision progressively decreases as k increases and that the critical parameter in the configuration is the overall k, rather than k_l and k_r.

We close this section with a note on the extension of our approach to loose associations among an arbitrary number of fragments. To build a loose association over fragments $\{F_1, \ldots, F_n\}$, a k_i-grouping is defined on each fragment F_i and the n-ary association is the table A with tuples $\{g_1, \ldots, g_n\}$ derived from the tuples in the original relation. The definitions of group association, alikeness, association heterogeneity, and deep heterogeneity have to be extended to take into consideration the fact that the association puts values of different fragments in relationship and that constraints may involve different sets of fragments. With respect to the protection degree, the loose association is guaranteed to be k-loose with any $k \leq min(k_i \cdot k_j)$ $\forall i, j = 1, \ldots, n,\ i \neq j$. Also, any binary association obtained projecting the n-ary association over the groupings of two fragments F_l and F_r will be guaranteed to be k-loose with any $k \leq k_l \cdot k_r$. Since experiments show that utility is correlated with the value of k, any query involving pairs of fragments will continue to enjoy the same utility as in the case of individual binary k-loose associations.

7. RELATED WORK

Several research efforts have addressed the problem of protecting privacy in data publication [4, 7, 8, 9, 10], also considering data utility (e.g., [6]). Among them, Anatomy [10] presents some similarities with our proposal. The tuples of the original relation are clustered in groups of ℓ tuples and a fragmentation is produced by splitting attributes between the quasi-identifier (on one side) and the sensitive attribute (on the other side) and reporting the group identifier in both fragments. While our proposal and Anatomy share the idea of fragmenting data and publishing associations at the group level, Anatomy considers only confidentiality constraints protecting the association of a single sensitive attribute with the respondents' quasi-identifier. Our work supports the presence of multiple of such attributes and, in general, it addresses a more complex scenario, accommodating generic confidentiality constraints and visibility requirements that capture the needs for data publication. Also, the use of two parameters instead of a single k or ℓ, grants us more flexibility in grouping. In scenarios where Anatomy is applicable, loose associations provide the same protection and utility guarantees as Anatomy (see Appendix B), which can then be considered a specific instance of our approach.

The use of data fragmentation to solve confidentiality constraints has been first proposed, in conjunction with encryption, in the context of data outsourcing [1, 2]. Our proposal, besides departing from the use of encryption, addresses a completely different problem, also introducing visibility requirements and loose associations.

The idea of considering associations among groups of tuples, in contrast to associations among tuples, has been first introduced in [4], where the authors aim at protecting many-to-many associations between two relations. Although our work and the proposal in [4] share the same high-level goal of protecting the sensitive associations, the two approaches are considerably different. First, we frame and solve a more general problem of data publishing where both confidentiality constraints and visibility requirements need to be respected. Also, in [4] duplicate values were not considered, thus leaving all the associations involving non-key attributes with multiple occurrences potentially exposed. Our approach considers the case of duplicate values and provides a general formalization of the problem, defining different heterogeneity properties which enjoy different levels of protection.

8. CONCLUSIONS

We presented an approach that opens a new direction for managing the problem of publishing data while respecting the privacy of sensitive information. Our approach enjoys large applicability, with the consideration of generic confidentiality constraints as well as visibility requirements expressing demands for data publication. We believe that fragmentation and loose associations can become an important modeling for the exploitation of the huge potential deriving from privacy-compliant access to large collections of data.

9. ACKNOWLEDGMENTS

This work was supported in part by the EU (project "PrimeLife", 216483); Italian MIUR (project "PEPPER" 2008SY2PH4); NSF (grants CT-20013A, CT-0716567, CT-0716323, CT-0627493, and CCF-1037987); AFOSR (grants FA9550-07-1-0527, FA9550-09-1-0421, and FA9550-08-1-0157); and ARO (DURIP award W911NF-09-01-0352).

10. REFERENCES

[1] G. Aggarwal et al. Two can keep a secret: a distributed architecture for secure database services. In *Proc. of CIDR*, USA, 2005.

[2] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Combining fragmentation and encryption to protect privacy in data storage. *ACM TISSEC*, 2010.

[3] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-Anonymity. In T. Yu and S. Jajodia, eds., *Secure Data Management in Decentralized Systems*. Springer-Verlag, 2007.

[4] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang. Anonymizing bipartite graph data using safe groupings. In *Proc. of VLDB*, New Zealand, 2008.

[5] H. Hacigümüs, B. Iyer, and S. Mehrotra. Providing database as a service. In *Proc. of ICDE*, USA, 2002.

[6] D. Kifer and J. Gehrke. Injecting utility into anonymized datasets. In *Proc. of SIGMOD*, USA, 2006.

[7] K. LeFevre, D. DeWitt., and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Proc. of ICDE*, USA, 2006.

[8] A. Machanavajjhala, J. Gehrke, and D. Kifer. ℓ-diversity: Privacy beyond k-anonymity. In *Proc. of ICDE*, USA, 2006.

[9] P. Samarati. Protecting respondents' identities in microdata release. *IEEE TKDE*, 13(6), 2001.

[10] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In *Proc. of VLDB*, Korea, 2006.

APPENDIX

A. COMPUTING A CORRECT AND MINIMAL FRAGMENTATION

The translation of Problem 4.1 into an instance of the SAT problem exploits the interpretation of the inputs to Problem 4.1 as boolean formulas. Visibility requirements are already represented as boolean formulas. Each confidentiality constraint can be represented with a boolean formula as a conjunction of the attributes appearing in the constraint. A possible fragmentation representing a solution to the problem can be interpreted as a truth assignment over boolean variables of the problem. Since a fragmentation corresponds to a set of fragments, each of which is a set of attributes, we need to distinguish the different fragments within the fragmentation (as we need different truth assignments for the different fragments). For each attribute of the original relation we therefore need to specify a different variable for each fragment in the fragmentation. More precisely, when considering m fragments, each attribute $a{\in}S$ will have m instances $a^1,\ldots,a^m$ characterizing its truth value in the different fragments. Confidentiality constraints, visibility requirements, and heterogeneity properties to be satisfied need to take into account these different instantiations. Given a confidentiality constraint c, or a visibility requirement v, we define the instantiation of c (v, resp.) with respect to F_i, denoted c^i (v^i, resp.), as the boolean formula c (v, resp.) where all the variables in c (v, resp.) are substituted by their i-th instantiation.

THEOREM A.1. *Given a relation schema $S(a_1,\ldots,a_n)$, a set $\mathcal{C}$ of confidentiality constraints over S, and a set $\mathcal{V}$ of visibility requirements over S, a correct fragmentation $\mathcal{F}$ composed of m fragments is a truth assignment for variables a_j^i, $j=1,\ldots,n, i=1,\ldots,m$, of the formula $\varphi^m=\varphi^m_{\mathcal{C}_{(1)}}\wedge\varphi^m_{\mathcal{C}_{(2)}}\wedge\varphi^m_{\mathcal{V}}\wedge\varphi^m_{\min}$ where:*

- $\varphi^m_{\mathcal{C}_{(1)}}=\bigwedge\limits_{j=1}^{|\mathcal{C}|}(\bigwedge\limits_{i=1}^{m}\neg c_j^i)$
- $\varphi^m_{\mathcal{C}_{(2)}}=\bigwedge\limits_{j=1}^{|S|}(\bigwedge\limits_{\substack{i,l=1\\ i\neq l}}^{m}\neg(a_j^i\wedge a_j^l))$
- $\varphi^m_{\mathcal{V}}=\bigwedge\limits_{j=1}^{|\mathcal{V}|}(\bigvee\limits_{i=1}^{m}v_j^i)$
- $\varphi^m_{\min}=\bigwedge\limits_{i=1}^{m}\neg a_j^i \qquad s.t.\ \nexists v\in\mathcal{V}, a_j\ in\ v$

PROOF. Omitted for space reasons. □

$$\varphi^1_{\mathcal{C}_{(1)}}=\neg\texttt{s}^1\wedge\neg(\texttt{p}^1\wedge\texttt{i}^1)\wedge\neg(\texttt{p}^1\wedge\texttt{d}^1)\wedge\neg(\texttt{b}^1\wedge\texttt{z}^1\wedge\texttt{i}^1)\wedge\neg(\texttt{b}^1\wedge\texttt{z}^1\wedge\texttt{d}^1)$$
$$\varphi^1_{\mathcal{C}_{(2)}}=\emptyset$$
$$\varphi^1_{\mathcal{V}}=(\texttt{p}^1\vee\texttt{z}^1)\wedge((\texttt{b}^1\wedge\texttt{z}^1)\vee\texttt{s}^1)\wedge(\texttt{i}^1\wedge\texttt{d}^1)$$
$$\varphi^1_{\min}=\emptyset$$

$$\varphi^2_{\mathcal{C}_{(1)}}=\neg\texttt{s}^1\wedge\neg\texttt{s}^2\wedge\neg(\texttt{p}^1\wedge\texttt{i}^1)\wedge\neg(\texttt{p}^2\wedge\texttt{i}^2)\wedge\neg(\texttt{p}^1\wedge\texttt{d}^1)\wedge\neg(\texttt{p}^2\wedge\texttt{d}^2)\wedge$$
$$\neg(\texttt{b}^1\wedge\texttt{z}^1\wedge\texttt{i}^1)\wedge\neg(\texttt{b}^2\wedge\texttt{z}^2\wedge\texttt{i}^2)\wedge\neg(\texttt{b}^1\wedge\texttt{z}^1\wedge\texttt{d}^1)\wedge\neg(\texttt{b}^2\wedge\texttt{z}^2\wedge\texttt{d}^2)$$
$$\varphi^2_{\mathcal{C}_{(2)}}=\neg(\texttt{p}^1\wedge\texttt{p}^2)\wedge\neg(\texttt{b}^1\wedge\texttt{b}^2)\wedge\neg(\texttt{z}^1\wedge\texttt{z}^2)\wedge\neg(\texttt{i}^1\wedge\texttt{i}^2)\wedge\neg(\texttt{d}^1\wedge\texttt{d}^2)$$
$$\varphi^2_{\mathcal{V}}=((\texttt{p}^1\vee\texttt{z}^1)\vee(\texttt{p}^2\vee\texttt{z}^2))\wedge(((\texttt{b}^1\wedge\texttt{z}^1)\vee\texttt{s}^1)\vee((\texttt{b}^2\wedge\texttt{z}^2)\vee\texttt{s}^2))\wedge$$
$$((\texttt{i}^1\wedge\texttt{d}^1)\vee(\texttt{i}^2\wedge\texttt{d}^2))$$
$$\varphi^2_{\min}=\emptyset$$

Figure 7: Computation of a fragmentation

A truth assignment (if it exists) for variables appearing in φ^m corresponds to a correct fragmentation composed of m fragments. Each component of conjunctive formula φ^m expresses conditions imposed by the input (confidentiality constraints and visibility requirements) and consequent properties to be guaranteed by the solution (fragmentation). Their semantics is as follows:

- $\varphi^m_{\mathcal{C}_{(1)}}$: every constraint must evaluate to false for every fragment (condition 1 of Definition 2.3);
- $\varphi^m_{\mathcal{C}_{(2)}}$: fragments should not have attributes in common, that is, every attribute must have at most one instance assuming a true value in the solution (condition 2 of Definition 2.3);
- $\varphi^m_{\mathcal{V}}$: every visibility requirement must be satisfied by some fragment (Definition 3.2);
- $\varphi^m_{\min}$: attributes not appearing in visibility requirements should remain false in the solution, that is, should not belong to any fragment. Note that this component of the formula is not needed for the correctness of the solution. It can be used to specify that attributes not explicitly mentioned in visibility requirements should not be released.

Theorem A.1 provides an instantiation of a SAT formula for finding a correct fragmentation (if it exists) composed of m fragments. Our approach for solving Problem 4.1 is to iterate the evaluation of a SAT solver, starting with one fragment ($m=1$) and increasing fragments by one at each iteration, until either a solution is found or m reaches the minimum among: the cardinality of the constraints, the cardinality of the visibility requirements, and the number of attributes appearing in visibility requirements. Such an approach retrieves a solution that has the *minimum number of fragments*. This approach has been implemented and showed significant performance (Appendix B).

EXAMPLE A.1. *Consider relation* HOSPITAL, *the confidentiality constraints, and the visibility requirements in Figure 1. Figure 7 illustrates the corresponding SAT formulation for m = 1 and m = 2 fragments (attributes are denoted by their initials). The SAT instance for one fragment is not satisfiable, while the SAT instance for two fragments is satisfied either by assigning* true *to* $\texttt{b}^1$, $\texttt{z}^1$, $\texttt{i}^2$, *and* $\texttt{d}^2$ *and* false *to all other variables, or by assigning* true *to* $\texttt{p}^1$, $\texttt{b}^1$, $\texttt{z}^1$, $\texttt{i}^2$, *and* $\texttt{d}^2$ *and* false *to all other variables. The two corresponding fragmentations are* $\mathcal{F}_1$={{Birth,ZIP},{Illness,Doctor}} (*Figure*1(d)) *and* $\mathcal{F}_2$={{Patient,Birth,ZIP},{Illness,Doctor}}.

If a fragmentation is minimal, merging any two of its fragments would violate at least one confidentiality constraint, as stated by the following theorem.

THEOREM A.2. *Given a set $\mathcal{C}$ and $\mathcal{V}$ of confidentiality constraints and visibility requirements over S, respectively, and a correct minimal fragmentation $\mathcal{F}$, $\forall F_l,F_r{\in}\mathcal{F}$, $F_l{\neq}F_r$, $\exists c{\in}\mathcal{C}{:}c{\subseteq}F_l{\cup}F_r$.*

PROOF. The proof is by contradiction. Suppose that $\nexists c\in\mathcal{C}{:}c\subseteq F_l\cup F_r$ and that $\mathcal{F}'{=}\mathcal{F}\setminus\{F_l,F_r\}\cup\{F_z\}$, with $F_z{=}F_l\cup F_r$. We prove that $\mathcal{F}'$ is a correct fragmentation.

- Since $\mathcal{F}$ is a safe fragmentation, $\forall F\in\mathcal{F}$, $\forall c\in\mathcal{C}$, $c\not\subseteq F$. Also, by assumption, $\nexists c\in\mathcal{C}{:}c\subseteq F_z$. Therefore $\mathcal{F}'$ satisfies the first condition in Definition 2.3.
- Since $\mathcal{F}$ is a safe fragmentation, $\forall F_i,F_j\in\mathcal{F}$, $i\neq j$: $F_i\cap F_j{=}\emptyset$. Therefore, $\forall F_i\in\mathcal{F}'$, $i\neq z$: $F_i\cap F_z{=}\emptyset$, since both $F_i\cap F_l{=}\emptyset$ and $F_i\cap F_r{=}\emptyset$. Then, $\mathcal{F}'$ satisfies the second condition in Definition 2.3.
- Since $\mathcal{F}$ satisfies all visibility requirements, $\forall v\in\mathcal{V}$, $\exists F\in\mathcal{F}{:}F\rightarrow v$. Since $F_z\rightarrow F_l$ and $F_z\rightarrow F_r$, $\forall v\in\mathcal{V}$: $F_l\rightarrow v\ \vee\ F_r\rightarrow v$, we have that $F_z\rightarrow v$. Therefore $\mathcal{F}'$ satisfies all visibility requirements (Definition 3.2).

We can then conclude that $\mathcal{F}'$ is a correct fragmentation such that $|\mathcal{F}'|{=}|\mathcal{F}|-1$. This implies that $\mathcal{F}$ is not a minimal fragmentation of S, contradicting the initial hypothesis. □

B. IMPLEMENTATION AND EXPERIMENTS

We implemented a prototype for the evaluation of the behavior of the techniques presented in the paper. The prototype is composed of two tools written in C/C++, implementing the two algorithms that solve Problem 4.1 and Problem 5.1, respectively. Experiments have been run on a PC with two Intel Xeon Quad 2.0GHz L3-4MB, 12GB RAM, four 1-Tbyte disks, and a Linux Ubuntu 9.04 operating system. The first tool solves Problem 4.1. It receives as input a relational schema, a set of confidentiality constraints, and a set of visibility requirements. It produces a solution to Problem 4.1 according to the translation into SAT instances illustrated in Appendix A and passes it to the Yices SAT solver (http://yices.csl.sri.com). We tested our tool with randomly generated configurations, observing significant performance. For instance, in less than 2 seconds the system was able to generate and solve the fragmentation for a configuration with 40 attributes and 16 among constraints and visibility requirements; the production of the 25,792 assertions to be passed as input to the SAT solver required 1980 ms, whereas the SAT solver identified a solution in 2 ms. Even considering much larger (probably unreal) configurations, the time taken by the SAT solver remained negligible. As an example, a configuration with 2500 attributes and 2000 among constraints and visibility requirements required 16 ms to be solved. The scalability of the tool provides confidence on the ability of the approach to manage large database schemas and complex privacy and visibility requirements.

The second tool solves Problem 5.1. It implements a greedy algorithm to determine a (k_l,k_r)-grouping, inducing a k-loose association, with $k{=}k_l{\cdot}k_r$. The algorithm first determines the maximum number of groups necessary for each fragment (i.e., $\lfloor|s|/k_l\rfloor$ for F_l and $\lfloor|s|/k_r\rfloor$ for F_r). It then scans all the tuples in the original relation s. For each tuple $t\in s$, the algorithm tries to place the corresponding right (left, resp.) sub-tuple r (l) in a group of the right (left) fragment that guarantees the satisfaction of Properties 5.1, 5.2, and 5.3, and that contains less than k_r (k_l) sub-tuples. We tested the precision of the queries executed on the fragments when a k-loose association is published. The experiments have been executed on both synthetic data sets and on the CENSUS data set (IPUMS-USA, http://www.ipums.org); we only report the results on the CENSUS data set since those performed on synthetic data sets show similar behaviors. We ran two sets of experiments: the first set compared our approach with Anatomy [10], to show how it exhibits the same behavior in the scenarios supported by Anatomy; the second set evaluated our approach with different values of k. The results illustrated in the following were computed as the average over 3 runs of the experiments, where each run considered 1000 different queries. The queries are SELECT FROM WHERE SQL queries, returning the result of the COUNT aggregation function over a subset of the tuples in the table. The WHERE clause is characterized by a condition of the form $\bigwedge_{i=1}^{n}(\bigvee_{j=1}^{m} a_i = v_{i_j})$, where a_i, $i=1,\ldots,n$, is an attribute in $F_l\cup F_r$ and v_{i_j}, $i=1,\ldots,n$ and $j=1,\ldots,m$, is a value in the domain of attribute a_i. The attributes and values populating the conditions in the WHERE clause of the queries have been randomly chosen.

The first set of experiments aimed at a direct comparison between the behavior of Anatomy and the use of a k-loose association induced by a $(1,k)$-grouping (in these experiments, k corresponds to parameter ℓ used in [10]). We retrieved from the Web site of the authors of Anatomy the program that was used to produce the experiments described in [10]. Our experiments, consistently with the code we retrieved, started from the CENSUS data set by choosing a subset of around 100,000 tuples (among the 500,000 in the data set), and then analyzed the behavior of Anatomy with $k=10$ and $k=12$. We considered a configuration with two fragments: one fragment contained the quasi-identifier attributes, and the other fragment contained the sensitive attribute. The error in the evaluation of the considered count queries showed a difference of less than 0.1% between the configurations obtained by [10] and those obtained by the use of our algorithm for loose associations. This difference can be explained as produced by the random generation of tuple groups, confirming the observation that Anatomy can be considered a specific instance of loose associations.

The second set of experiments aimed at evaluating how the error in the evaluation of queries evolves with the increase in the value of k in our approach. We were also interested in analyzing, for each k, the behavior of the technique for configurations with varying values of k_l and k_r (always having $k_l\cdot k_r=k$). The set of constraints we used in these experiments is more extensive than that used for the previous set of experiments, to better solicit the heterogeneity properties. These configurations are not supported by Anatomy. Figure 8 shows the results of the experiments for configurations with k varying from 10 to 20, with the following configurations: (1,10), (2,5); (1,12), (2,6), (3,4); (1,14), (2,7); (1,16), (2,8), (4,4); (1,18), (2,9), (3,6); (1,20), (2,10), (4,5). The experiments confirm two important aspects of our technique. First, the level of precision progressively decreases as k increases. The point associated with each value k represents the average of the experiments for the different configurations; the error presents a regular increase. Second, the critical parameter in the configuration is the overall privacy degree k, rather than the values of terms k_l and k_r. We can observe that the estimates for configurations with the same k value typically lie close to each other. Across all values of k, the precision of the queries on the fragments mostly depends on the product of k_l and k_r, rather than on the relative sizes of k_l and k_r. Hence, configurations with both k_l and k_r greater than 1 produce

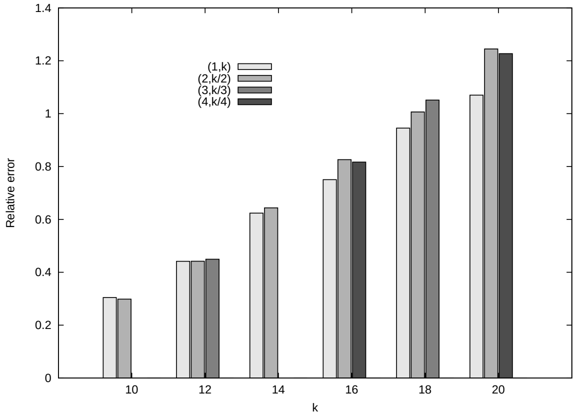


Figure 8: Error obtained varying (k_l,k_r)

utility similar to that of (1,k) configurations, while offering greater flexibility in their computation (Theorem 5.3).

C. PROOFS

Theorem 4.1. *The* Min-CF *problem is NP-hard.*

Proof. The proof is a reduction from the NP-hard problem of minimum hypergraph coloring, formulated as follows: *given a hypergraph $\mathcal{H}(N,E)$, determine a minimum coloring of $\mathcal{H}$, that is, assign to each vertex in N a color such that adjacent vertices have different colors, and the number of colors is minimized.*

Given a relation schema S, a set $\mathcal{C}$ of confidentiality constraints, and a set $\mathcal{V}$ of visibility requirements, the correspondence between the Min-CF problem and the hypergraph coloring problem can be defined as follows. Any vertex n_i of hypergraph $\mathcal{H}$ translates to an attribute $a{\in}S$ and to a visibility requirement $v{=}a{\in}\mathcal{V}$. Any edge e_i in $\mathcal{H}$, which connects $n_{i_1},\ldots,n_{i_c}$, translates to a (non singleton) constraint $c_i{=}\{a_{i_1},\ldots,a_{i_c}\}$ in $\mathcal{C}$. A fragmentation $\mathcal{F}{=}\{F_1(a_{1_1},\ldots,a_{1_k}),\ldots,F_m(a_{m_1},\ldots,a_{m_l})\}$ of S satisfying all constraints in $\mathcal{C}$ and all visibility requirements in $\mathcal{V}$ corresponds to a solution *Sol* for the corresponding hypergraph coloring problem. Specifically, *Sol* uses m colors and all nodes $n_{j_{i_1}},\ldots,n_{j_{i_n}}$, corresponding to the attributes in F_j, $j=1,\ldots,m$, are colored using the j-th color. Hence, the Min-CF problem is NP-hard. □

Theorem 5.1. *The* Min k-loose *problem is NP-hard.*

Proof. The proof is a reduction from the NP-hard problem of maximum 3-dimensional matching, formulated as follows: *given a set $T\subseteq X{\times}Y{\times}Z$, where X, Y, Z are disjoint sets, determine a matching $M\subseteq T$ of maximum cardinality, such that no elements in M agree in any coordinate.*

We prove that a (k_l,k_r)-grouping of an instance s of relation S corresponding to T and inducing a k-loose association, with $k{\leq}k_l{\cdot}k_r$, is a matching of size $k{-}1$ for T. Consider relation schema $S(\underline{id},a_x,a_y,a_z,a^d_1,\ldots,a^d_n)$, with $n{=}k(m-k)$ and $m=|T|$; a set $\mathcal{C}$ of confidentiality constraints, where $\forall a{\in}S$, with $a{\neq}id$, $c{=}\{id,a\}{\in}\mathcal{C}$; a fragmentation $\mathcal{F}$ composed of two fragments: $F_l{=}\{id\}$ and $F_r{=}\{a_x,a_y,a_z,a^d_1,\ldots,a^d_n\}$; and an instance s of relation S defined as follows:

- *Real tuples:* $t_1,\ldots,t_m$, with $m=|T|$. $\forall i{=}1,\ldots,m$, $t_i[a_x]$, $t_i[a_y]$, $t_i[a_z]$ represent the content of the considered set T, that is, $\langle t_i[a_x],t_i[a_y],t_i[a_z]\rangle{\in}T$; $t_i[id]{=}i$; and $t_i[a^d_j]{=}{-}i$, $\forall j{=}1,\ldots,n$. Then, $\forall t_i,t_j{\in}s$, with $i{\neq}j$, $t_i{\simeq}t_j$ iff either $t_i[a_x]{=}t_j[a_x]$, or $t_i[a_y]{=}t_j[a_y]$, or $t_i[a_z]{=}t_j[a_z]$.
- *Dummy tuples:* $t^d_1,\ldots,t^d_n$, with $n{=}k(m-k)$. $\forall i{=}1,\ldots,n$, $t^d_i[a_x]{=}\max(X){+}i$; $t^d_i[a_y]{=}\max(Y){+}i$; $t^d_i[a_z]{=}\max(Z){+}i$, where $\max(X)$ ($\max(Y)$ and $\max(Z)$, resp.) represents the maximum value for X (Y and Z, resp.) in T; $t^d_i[id]{=}m{+}i$; and $t^d_i[a^d_j]{=}i$, $\forall j{=}1,\ldots,n$. Then, $\forall t^d_i,t^d_j{\in}s$ with $i{\neq}j$, $t^d_i{\not\simeq}t^d_j$, since t^d_i and t^d_j have different values on all attributes. Also, $\forall t_i,t^d_j{\in}s$, $t_i{\not\simeq}t^d_j$, since t_i and t^d_j have different values on all attributes.
- *Star tuple:* t^*. $t^*[a_x]{=}\max(X){+}n{+}1$, $t^*[a_y]{=}\max(Y){+}n{+}1$, $t^*[a_z]{=}\max(Z){+}n{+}1$; $t^*[id]{=}m{+}n{+}1$; and $t^*[a^d_j]{=}j$, $\forall j{=}1,\ldots,n$. Then, $\forall t^d_i{\in}s$, $t^*{\simeq}t^d_i$, since they have the same value for an attribute $t^*[a^d_i]{=}t^d_i[a^d_i]{=}i$ appearing in a confidentiality constraint $c=\{id,a^d_i\}$; $\forall t_i{\in}s$, $t^*{\not\simeq}t_i$, since t^* and t_i have different values on all attributes.

Let $\mathcal{G}_l$ and $\mathcal{G}_r$ be a (k_l,k_r)-grouping of s inducing a k-loose association A. The set of tuples $T^*{=}\bigcup_j\{\mathcal{G}_r^{-1}(g_{r_j})$: $(g_l,g_{r_j}){\in}A$, $g_l{=}\mathcal{G}_l(l^*)\}$, where $l^*=t^*[F_l]$, must be a subset of the set $\{r^*,r_1,\ldots,r_m\}$ of tuples corresponding to the projection of the real tuples $\{t^*,t_1,\ldots,t_m\}$ on the attributes in F_r and must contain at least k items, since otherwise A would not be k-loose (Definition 5.1). Also, $\forall t_i,t_j{\in}T^*$, with $i{\neq}j$, $t_i{\not\simeq}t_j$ and therefore all the tuples in T^* have different values on attributes a_x, a_y, and a_z. Therefore, the set $T^*{\setminus}\{r^*\}$ of real tuples projected on a_x, a_y, and a_z represents a matching of size $k{-}1$ for T. Note that if a k-loose association does not exist for s, then there does not exist a matching of size $k{-}1$ for T. To compute maximal matching for T, it is necessary to compute a k-loose association, if it exists, iteratively decreasing k from $m+1$ to 2, until a k-loose association is found. As a consequence, the Min k-loose problem is NP-hard. □

Theorem 5.2. *Given a set $\mathcal{C}$ of confidentiality constraints, a fragmentation $\mathcal{F}$ of S, relation s over S, two fragments F_l and F_r in $\mathcal{F}$, their instances f_l and f_r, and a minimal (k_l,k_r)-grouping that satisfies Properties 5.1, 5.2, and 5.3, then the group association A induced by the (k_l,k_r)-grouping is k-loose (Definition 5.4) for each $k{\leq}k_l{\cdot}k_r$, and is minimal k-loose for $k{=}k_l{\cdot}k_r$.*

Proof. By Definition 5.1, each group $g_i{\in}\mathsf{GID}_l$ contains at least k_l tuples and each group $g_j{\in}\mathsf{GID}_r$ contains at least k_r tuples. Hence, Property 5.2 implies that: *1)* each group $g_i{\in}\mathsf{GID}_l$ is associated with at least k_l different groups in GID_r, denoted *groups_rhs$_i$*, and *2)* each group $g_j{\in}\mathsf{GID}_r$ is associated with at least k_r different groups in GID_l, denoted *groups_lhs$_j$*. Properties 5.1 and 5.3 guarantee that *groups_rhs$_i$* and *groups_lhs$_j$* do not contain tuples that are alike. As a consequence, each $g_i{\in}\mathsf{GID}_l$ is associated with a total number of tuples in f_r greater than or equal to $|$*groups_rhs$_i$* $|{\cdot}k_r{\geq}k_l{\cdot}k_r$, none of which are alike. Analogously, each group $g_j{\in}$ GID_r is associated with a total number of tuples in f_l greater than or equal to $k_l{\cdot}|$*groups_lhs$_j$* $|{\geq}k_l{\cdot}k_r$, none of which are alike. Then, the

(k_l,k_r)-grouping satisfying Properties 5.1, 5.2, and 5.3 induces a group association that is k-loose for each $k{\leq}k_l{\cdot}k_r$. Suppose that $k{=}k_l{\cdot}k_r$ and that the (k_l,k_r)-grouping does not induce a minimal k-loose association. Hence, there exists a (k'_l,k'_r)-grouping such that $k{\leq}k'_l{\cdot}k'_r{<}k_l{\cdot}k_r{=}k$, which is a contradiction. Therefore, the (k_l,k_r)-grouping induces a minimal $k_l{\cdot}k_r$-loose association. □

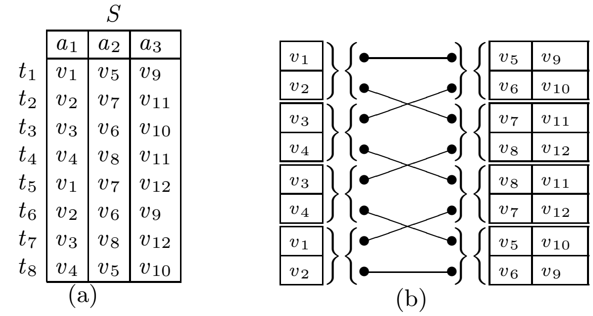
THEOREM 5.3. *Given two fragments F_l and F_r in $\mathcal{F}$, their instances f_l and f_r, a privacy degree k, and a number n of constraints c such that $c{\subseteq}F_l{\cup}F_r$, then*

1. *if $n = 1$: $\exists$ a flat grouping providing k-looseness $\Longleftrightarrow$ $\exists$ a sparse grouping providing k-looseness;*
2. *if $n > 1$:*
   (a) *$\exists$ a flat grouping providing k-looseness $\Longrightarrow$ $\exists$ a sparse grouping providing k-looseness;*
   (b) *$\exists$ a sparse grouping providing k-looseness $\not\Longleftarrow$ $\exists$ a flat grouping providing k-looseness.*

PROOF. (SKETCH) Due to space constraints, we provide only a sketch of the proof, which is organized in three steps. **Step 1:** *if $n{\geq}1$: $\exists$ a flat grouping providing k-looseness $\Longrightarrow$ $\exists$ a sparse grouping providing k-looseness.* Consider a flat (1,k)-grouping $\mathcal{G}_l$, $\mathcal{G}_r$ (the case of (k,1)-grouping is symmetric), with $k_l{\cdot}k_r{=}k$ that provides k-looseness. We show that this grouping can be transformed in a minimal (k_l,k_r)-grouping providing k-looseness. For simplicity and without loss of generality, we suppose that $|s|$ mod $k{=}0$, where s is the original relation. The basic idea is that the tuples in $\mathcal{G}_r^{-1}(g_r)$, which are exactly k, can be partitioned into k_l different groups, containing exactly k_r tuples. The k_r-grouping of f_r induces a k_l-grouping of f_l, where each group contains k_l non-alike tuples. By construction, the (k_l,k_r)-grouping satisfies Properties 5.1, 5.2, and 5.3.
Step 2: *if $n{=}1$: $\exists$ a sparse grouping providing k-looseness $\Longrightarrow$ $\exists$ a flat grouping providing k-looseness.* The proof considers two cases.

a) We prove that if $\exists l{\in}f_l{:}|\{l_i|l_i{\in}f_l,\ l{\simeq}l_i\}| > \frac{|s|}{k}$ or if $\exists r{\in}f_r{:}|\{r_i|r_i{\in}f_r,\ r{\simeq}r_i\}| > \frac{|s|}{k}$, than $\nexists$ either a flat or a sparse grouping providing k-looseness. Suppose that $\exists l{\in}f_l{:}|\{l_i|l_i{\in}f_l,\ l{\simeq}l_i\}| > \frac{|s|}{k}$ (the case of $r{\in}f_r$ is symmetric). It is easy to see that a (k,1)-grouping violates Property 5.1 and that a (1,k)-grouping violates Property 5.3. If there exists a (k_l,k_r)-grouping providing k-looseness, then $|\mathsf{GID}_r|{\geq}(\frac{|s|}{k}+1)\cdot k_l$ and $|s|{=}(\frac{|s|}{k}+1)\cdot k_l\cdot k_r$, thus bringing to a contradiction.

b) We prove that if $\forall t{\in}f_l$: $|\{t_i|t_i{\in}f_l,\ t{\simeq}t_i\}| \leq \frac{|s|}{k}$ and $\forall t{\in}f_r$: $|\{t_i|t_i{\in}f_r,\ t{\simeq}t_i\}| \leq \frac{|s|}{k}$, then $\exists$ both a flat and (by Step 1) a sparse grouping providing k-looseness. For simplicity, suppose $F_l[c]{=}a_l$ and $F_r[c]{=}a_r$. Consider a (k,1)-grouping $\mathcal{G}_l$, $\mathcal{G}_r$, with $\mathsf{GID}_l{=}\{g_{l_1},\ldots,g_{l_h}\}$, satisfying Properties 5.1, 5.2, and 5.3. Let $|\mathcal{G}_l^{-1}(g_{l_i})| = k$, $i = 1,\ldots,(h-1)$, and $|\mathcal{G}_l^{-1}(g_{l_h})| = k-1$. Tuple t is the unique tuple such that $\mathcal{G}_l(t[F_l]){=}\mathcal{G}_r(t[F_r]){=}\textsc{null}$, where $t[a_l]{=}l$ and $t[a_r]{=}r$. Both l and r appear $\frac{|s|}{k}$ times in f_l and f_r, respectively. We need to prove that it is always possible to define $\mathcal{G}_l(t[F_l])$ and $\mathcal{G}_r(t[F_r])$ guaranteing k-looseness. Note that there exists $g_{l_i}{\in}\mathsf{GID}_l$ such that by assigning $\mathcal{G}_l(t[F_l]){=}g_{l_i}$ Property 5.1 is satisfied, and there exists $g_{l_j}{\in}\mathsf{GID}_l$ such that by assigning $\mathcal{G}_l(t[F_l]){=}g_{l_j}$ Property 5.3 is satisfied. Four different cases may occur:



S

	a_1	a_2	a_3
t_1	v_1	v_5	v_9
t_2	v_2	v_7	v_{11}
t_3	v_3	v_6	v_{10}
t_4	v_4	v_8	v_{11}
t_5	v_1	v_7	v_{12}
t_6	v_2	v_6	v_9
t_7	v_3	v_8	v_{12}
t_8	v_4	v_5	v_{10}

Figure 9: A relation s (a) and a (2,2)-grouping providing 4-looseness (b)

- $g_{l_i}{=}g_{l_j}{=}g_{l_h}$. By assigning $\mathcal{G}_l(t[F_l]){=}g_{l_h}$, k-looseness is satisfied.
- $g_{l_i}{=}g_{l_j}{\neq}g_{l_h}$. By assigning $\mathcal{G}_l(t[F_l]){=}g_{l_i}$, if $\exists t_i{\in}f_l$: $t_i[a_l]{=}l$ and $\mathcal{G}_l(t_i){=}g_{l_i}$, Property 5.1 is violated and the group association obtained does not satisfy k-looseness. However, there are always at least two tuples t_x,$t_y{\in}\mathcal{G}_l^{-1}(g_{l_i})$ such that, by setting $\mathcal{G}_l(t_x)$ and $\mathcal{G}_l(t_y)$ to g_{l_h}, Property 5.1 is not violated. Analogously, there are always at least two tuples t_w,$t_z{\in}\mathcal{G}_l^{-1}(g_{l_i})$ such that, by setting $\mathcal{G}_l(t_w)$ and $\mathcal{G}_l(t_z)$ to g_{l_h}, Property 5.3 is not violated. If w is equal to x (y, resp.) or z is equal to x (y, resp.), it is sufficient to set $\mathcal{G}_l(t_x)$ ($\mathcal{G}_l(t_y)$, resp.) to g_{l_h} to satisfy k-looseness. Otherwise, it is necessary to iteratively swap tuples between $\mathcal{G}_l^{-1}(g_{l_i})$ and $\mathcal{G}_l^{-1}(g_{l_h})$, starting from one among t_x, t_y, t_w, and t_z. At each step, the tuple inserted in a group might violate Property 5.1 or Property 5.3 and is therefore moved to another group. Since initially $\mathcal{G}_l$ and $\mathcal{G}_r$ satisfy Properties 5.1 and 5.3, at each insertion of a tuple in a group, at most another tuple needs to be moved. Since each tuple is moved to the same group no more than once, the process terminates, guaranteeing k-looseness.
- $g_{l_i}{=}g_{l_h}{\neq}g_{l_j}$ ($g_{l_j}{=}g_{l_h}{\neq}g_{l_i}$ is symmetric). There is already a tuple t_i in $\mathcal{G}_l^{-1}(g_{l_h})$ such that $t_i[a_r]{=}r$. Therefore, we apply the swapping process of tuples described above between $\mathcal{G}_l^{-1}(g_{l_h})$ and $\mathcal{G}_l^{-1}(g_{l_j})$, starting from t_i. Since $\mathcal{G}_l^{-1}(g_{l_j})$ contains k tuples while $\mathcal{G}_l^{-1}(g_{l_h})$ contains $k{-}1$ tuples, the process terminates. Also, the tuple t_i such that $t_i[a_l]{=}l$ cannot be moved to $\mathcal{G}_l^{-1}(g_{l_h})$ with this process. Therefore, by assigning $\mathcal{G}_l(t[F_l]){=}g_{l_h}$, k-looseness is satisfied.
- $g_{l_i}{\neq}g_{l_j}{\neq}g_{l_h}$. This situation can be reduced to the previous case, by first applying the swapping process of tuples between $\mathcal{G}_l^{-1}(g_{l_h})$ and $\mathcal{G}_l^{-1}(g_{l_j})$, starting from the tuple t_i in $\mathcal{G}_l^{-1}(g_{l_h})$ with $t_i[a_r]{=}r$, and then by applying the same process between $\mathcal{G}_l^{-1}(g_{l_h})$ and $\mathcal{G}_l^{-1}(g_{l_i})$, starting from the tuple t_i in $\mathcal{G}_l^{-1}(g_{l_h})$ with $t_i[a_l]{=}l$. By assigning $\mathcal{G}_l(t[F_l]){=}g_{l_h}$, k-looseness is satisfied.

Step 3: *if $n{>}1$: $\exists$ a sparse grouping providing k-looseness $\not\Longrightarrow$ $\exists$ a flat grouping providing k-looseness.* We provide a counterexample. Consider relation s in Figure 9(a) and suppose that $F_r{=}\{a_1\}$ and $F_l{=}\{a_2{,}a_3\}$, and $c_1{=}\{a_1{,}a_2\}$, $c_2{=}\{a_1{,}a_3\}$ are two confidentiality constraints. Figure 9(b) illustrates a (2,2)-grouping providing 4-looseness. Any minimal (4,1)-grouping defined over F_r and F_l violates Property 5.3, hence the implication does not hold. □