Inferences from Attribute-Disjoint and Duplicate-Preserving Relational Fragmentations

Joachim Biskup and Marcel Preuß

Fakultät für Informatik, Technische Universität Dortmund, Germany
{joachim.biskup|marcel.preuss}@cs.tu-dortmund.de

Abstract. The transmission of own and partly confidential data to another agent, e.g., for cloud computing, comes along with the risk of enabling the receiver to infer information he is not entitled to learn. We consider a specific countermeasure against unwanted inferences about associations between data values whose combination of attributes are declared to be sensitive. This countermeasure fragments a relation instance into attribute-disjoint and duplicate-preserving projections such that no sensitive attribute combination is contained in any projection. Though attribute-disjointness is intended to make a reconstruction of original data impossible for the receiver, the goal of inference-proofness will not always be accomplished. In particular, inferences might be based on combinatorial effects, since duplicate-preservation implies that the frequencies of value associations in visible projections equals those in the original relation instance. Moreover, the receiver might exploit functional dependencies, numerical dependencies and tuple-generating dependencies, as presumably known from the underlying database schema. We identify several conditions for a fragmentation to violate inference-proofness. Besides complementing classical results about lossless decompositions, our results could be employed for designing better countermeasures.

Keywords: Attribute-disjointness · Cloud computing · Database relation · Confidentiality · Duplicate-preservation · Fragmentation · Frequencies · Functional dependency · Inference-proofness · Numerical dependency · Projection · Sensitive association · Tuple-generating dependency

1 Introduction

A data owner might consider to somehow fragment his relational data and to only make the resulting fragments accessible to another agent, which, for a prominent example, might offer some cloud services to the owner. Such a fragmentation then aims at hiding some information about sensitive associations contained in the original data to the service agent. Thus, though in principle being seen as cooperating, the service agent is also perceived as potentially attacking the confidentiality interests of the owner by attempting to infer hidden original information from accessible data and, if applicable, additional background knowledge. Accordingly, the data owner should carefully choose a fragmentation technique and thoroughly investigate whether the resulting fragmentation of his specific data sufficiently satisfies his confidentiality interests.

Our considerations are motivated by the particular proposal of "combining fragmentation and encryption to protect privacy in data storage" [14], a technique which converts a given relation instance and some confidentiality requirements on the schema level into a set of vertical relational fragments all of which might be accessible for an attacker. We focus on three aspects of this proposal:

- The resulting fragmentation is *attribute-disjoint*, i.e., fragments do not share attributes and thus seem to be unrelated. Moreover, regarding internal storage and external display, the sequence of subtuple instances in a fragment is supposed to be fully independent of the sequences in other fragments, and of any sequence of tuple instances in the hidden data as well.
- Each fragment is *duplicate-preserving* and thus, for any values under attributes in the fragment, their frequency (i.e., number of occurrences) in the fragment is equal to their frequency in the hidden underlying relation instance.
- The attacker might see *all* fragments, and thus he is supposed to take advantage of knowing several views on the same hidden data.

Focusing on the enforcement of confidentiality requirements by means of fragmentation, we will purposely ignore all cryptographic aspects and neglect the details of reconstructability of the original data by the data owner. For further simplifying our investigations, we will also assume that none of the attributes get encrypted values:

- The fragmentation is *full*, i.e., it covers all attributes of the original relation.

For this setting, we will discuss various kinds of *successful inference attacks* based on observable *frequencies* of visible data items and on additional background knowledge in the form of *data dependencies* and actual *content data*, in spite of the *attribute-disjointness* at first glance generating unrelated fragments. In doing so, we will present some fundamental assertions about such inferences, together with some complexity considerations. The resulting main contribution will be the identification of both the crucial role of frequencies and the challenge to future research how to block their exploitation.

Example 1 (Fragmentation with encryption). This example illustrates the techniques proposed by V. Ciriani et al [14] by means of a simple relational schema *Patient* providing attributes ACTION, S(*ocial*)S(*ecurity*)N(*umber*), (*Patient*)NAME, ILLNESS, (*Prescribed*) MEDICATION, HURTBY, and (*Treating*) DOCTOR to record a unique tuple instance for each medical action. Figure 1 shows a relation instance containing 4 tuple instances.
Suppose that the owner wants to hide values of the singleton attribute set {SSN}, and value combinations for associations expressed by the non-singleton attribute sets {NAME, DOCTOR}, {NAME, MEDICATION}, {NAME, HURTBY}, and {ILLNESS, HURTBY}, respectively. Single values can only be protected by encryption. But value combinations of a sensitive association can be handled by

Patient	Action	SSN	Name	Illness	Medication	HurtBy	Doctor
	r_1	1234	Hellmann	Borderline	MedA	Hellmann	White
	r_2	2345	Dooley	Laceration	MedB	McKinley	Waren
	r_3	3456	McKinley	Laceration	MedB	Dooley	Waren
	r_4	3456	McKinley	Concussion	MedC	Dooley	Waren

Fig. 1. A relation instance for the relational schema *Patient*

F_1	Action	Name
	r_2	Dooley
	r_4	McKinley
	r_3	McKinley
	r_1	Hellmann

F_2	Illness	Doctor
	Laceration	Waren
	Borderline	White
	Concussion	Waren
	Laceration	Waren

F_3	Medication	HurtBy
	MedC	Dooley
	MedB	Dooley
	MedA	Hellmann
	MedB	McKinley

Fig. 2. A simplified fragmentation (without encryption related parts needed for reconstruction by the owner) of the relation instance of the schema *Patient*

fragmentation, i.e., by distributing the values occurring in an association among different fragments obtained by projections without duplicate removal, under the condition that the fragments do not overlap and, thus, are not obviously linked. One possible option is to partition the attributes of the schema – in this example except SSN – into the mutually disjoint sets {Action, Name}, {Illness, Doctor}, and {Medication, HurtBy}. Then, for each of them a fragment is generated that makes the values of the attribute set visible and stores the encryption of all remaining values under a new attribute, say Enc. In principle, but no longer considered in the remainder, we would have to manage the encrypted parts to enable the data owner to reconstruct the original tuple instances. And we have to suitably scramble the (sub)tuple instances generated for a fragment to block inferences based on their sequence displayed. A possible result, simplified as indicated, is shown in Figure 2.

Unfortunately, however, though often being helpful, in this example the attribute-disjointness does not guarantee the confidentiality requirements. In fact, while the fragment instances to F_1 and F_2 contain 4 subtuple instances each, there are 3 occurrences of the value “Waren” under the attribute Doctor for F_2 but only 2 occurrences of a value *different* from “McKinley” under the attribute Name for F_1. Hence, any matching of the fragment instances must combine at least one of the occurrences of “Waren” with an occurrence of “McKinley”. Thus, exploiting the visible frequencies of occurrences in the fragmentation, the occurrence of the value combination (McKinley,Waren) under the attribute combination {Name, Doctor} in the hidden original relation instance is inferrable, violating the confidentiality requirements. In general, besides frequencies we would also have to consider the impact of data dependencies.

After briefly introducing basic definitions in Section 2, we will present and discuss fundamental risks of harmful inferences by exploiting first only observable frequencies (Section 3) and subsequently additionally background knowledge in the form of data dependencies, in turn inspecting functional dependencies and more general numerical dependencies (Section 4) and finally tuple-generating dependencies with a multivalued dependency as a special case (Section 5). In the concluding Section 6, we point to related work, summarize our achievements, outline future research on blocking inferences of the kind treated, and highlight the connection of our study with the broader topics of inversion of database queries and reasoning under uncertainty.

2 Basic Definitions

Abstracting from any application, ignoring encryption related parts and the owner's need for reconstruction, and assuming the covering of all attributes, our investigations will treat fragmentations of the kind defined below using standard terminology [1], together with their impact on the protection of associations.

Definition 1 (Full attribute-disjoint and duplicate-preserving fragmentation). *Let* $(R(X), \mathcal{SC})$ *be a relational* schema *with attribute set* X *and data dependencies* $\mathcal{SC}$*, and* $\mathcal{X} = \langle X_1, \ldots, X_m \rangle$ *be a sequence of attribute sets partitioning* X*, i.e., the sets* X_i *are nonempty and mutually disjoint subsets of* X *such that* $X = \bigcup_{i=1,\ldots,m} X_i$*. Then* $\mathcal{F} = \langle F_1(X_1), \ldots, F_m(X_m) \rangle$ *is the* fragmentation schema *derived from* $R(X)$ *and* $\mathcal{X}$*.*

Furthermore, let r *be a relation* instance *of* $(R(X), \mathcal{SC})$*, i.e., a finite set of tuples (without duplicates) over the attributes in* X *satisfying all data dependencies in* $\mathcal{SC}$*, containing* n *different tuples. Then, seen as an operator, the fragmentation schema* $\mathcal{F}$ *generates the* fragmentation instance $\mathcal{F}(r) = \langle f_1, \ldots, f_m \rangle$ *by taking the projections of* r *on* X_i*, respectively,* without removing duplicates *and then* probabilistically scrambling *the order of them regarding storage or display,* $f_i = \bar{\pi}^{?}_{X_i}(r)$*, such that each fragment instance* f_i *has* n *subtuple instances*[1]*.*

We emphasize that the setting of Definition 1 requires

- the *absence of duplicates* in original relation instances r (meant to be actually stored under the relational schema $(R(X), \mathcal{SC})$ on the one hand, and
- the *suppression of duplicate removal* when generating the fragmentation instances by taking projections according to the fragmentation schema on the other hand.

While the duplicate preservation under fragmentation is essential for the techniques proposed by V. Ciriani et al [14], in an alternative approach, technically, we could allow duplicates already in original relation instances. However, most practical applications and both constraint-enforcement and query-answering based on first-order logic usually assume set semantics rather than multiset semantics for original relation instances and, accordingly, so do we.

[1] Where appropriate and convenient, we distinguish between a *tuple* and a *tuple instance*: we call an *assignment* of values to some attributes a tuple, whereas we refer to an *occurrence* of a tuple as a tuple instance having in mind that a relation instance allowing duplicates might contain multiple instances of the same tuple.

Definition 2 (Syntactically protected association). *Let $\mathcal{F} = \langle F_1(X_1), \ldots, F_m(X_m)\rangle$ be the fragmentation schema derived from a relational schema $(R(X), \mathcal{SC})$ and a sequence $\mathcal{X}$ of attribute sets partitioning X. Then an attribute set C is an* association syntactically protected *by $\mathcal{F}$ iff C is a non-singleton subset of X but not contained in any of the attribute sets X_i.*

Unfortunately, as already mentioned before, the syntactic splitting condition of Definition 2 might fail to ensure strong versions of confidentiality. In particular, an actually occurring value combination of an only syntactically protected association might be inferrable by means of considering so-called matchings.

Definition 3 (Matching-inferrable value combination). *Let $\mathcal{F} = \langle F_1(X_1), \ldots, F_m(X_m)\rangle$ be the fragmentation schema derived from a relational schema $(R(X), \mathcal{SC})$ and a sequence $\mathcal{X}$ of attribute sets partitioning X. Let $\mathcal{F}(r) = f = \langle f_1, \ldots, f_m\rangle$ be the fragmentation instance generated from the relation instance r of $(R(X), \mathcal{SC})$. Furthermore, let attribute set C be an association syntactically protected by $\mathcal{F}$. A subtuple μ over C is called a* matching-inferrable value combination *iff it is generated by* each *$\mathcal{SC}$-admissible matching M of the subtuples in f.*

Here, a matching *is formed by iteratively taking one subtuple instance from each fragment instance f_i and combining them until the fragment instances (all having the same number n of subtuple instances) are (simultaneously) exhausted. In this way, a matching M generates a collection $M(r)$ of n tuple instances – possibly containing duplicates – over the attributes of X. Moreover, a matching M is called $\mathcal{SC}$-admissible if $M(r)$ is an instance of $(R(X), \mathcal{SC})$, i.e., a set (containing no duplicates) satisfying all data dependencies in $\mathcal{SC}$.*

Remark 1. From an attacking observer's point of view, a matching M can be seen as *one* possibility to undo the unknown scrambling of subtuple instances when the fragment instances have been generated. Some possibilities, however, might produce duplicates or result in a violation of data dependencies, and thus have to be discarded. Accordingly, if an attacker can find out that a subtuple μ over an attribute set C is generated by *all* remaining $\mathcal{SC}$-admissible possibilities, he can conclude that this subtuple can be obtained by undoing the actually employed scrambling and, thus, occurs in the hidden relation instance. Hence, for a deliberately syntactically protected association C, such a subtuple would be matching-inferrable: a successful inference from an attacker's point of view, but a security violation from the owner's point of view.

Remark 2. More formally, if C is a deliberately syntactically protected association, then an attacking receiver is suspected to be interested in the *certain* part (formalized by *intersection*) of the *projections* on C of the *relation instances* r' of the *schema* $(R(X), \mathcal{SC})$ contained in the *inversion* of the observed *fragmentation instance* $f = \mathcal{F}(r)$ under the *fragmentation schema* $\mathcal{F}$, i.e., to determine

$$\mathcal{F}^{-1,C}_{\mathcal{SC}}(f) = \bigcap \{\, \pi_C(r') \mid r' \text{ is relation instance of } (R(X), \mathcal{SC}) \text{ and } \mathcal{F}(r') = f\}.$$

For the attacker, a conceptual solution is given by the matching procedure sketched above, due to the straightforward equation

$$\mathcal{F}^{-1,C}_{\mathcal{SC}}(f) = \bigcap \{\, \pi_C(M(r)) \mid M(r) \text{ is formed from } \mathcal{F}(r) = f \text{ and } \mathcal{SC}\text{-admissible} \,\}.$$

In contrast, the data owner would aim at assuring that the visible fragmentation f does not allow any possibilistic inference, i.e., that $\mathcal{F}^{-1,C}_{\mathcal{SC}}(f) = \emptyset$.

Remark 3. The data owner's goal to ensure $\mathcal{F}^{-1,C}_{\mathcal{SC}}(f) = \emptyset$ is also equivalent to the notion of *inference-proofness* as employed by the concept of Controlled Interaction Execution [6], under a confidentiality policy suitably expressing the need to hide value combinations over C, as elaborated in [11,10]. Though without referring to particular formal logic, and similarly as in an abstract version of Controlled Interaction Execution [7], the goal roughly says that a suitable logic-based formalization of the setting does not entail any sentence that logically expresses the subtuple μ over C.

3 Frequency-Based Inferences without Dependencies

To start with, we briefly remind a classical result of the theory of relational databases, see [1], that vertically decomposing a relation instance r – *without* duplicates – into covering projections $\pi_{X_i}(r)$ for $i = 1, \ldots, m$ – while *removing* duplicates – might be *lossy*, i.e., the (natural) join $\bowtie_{j=1\ldots m} \pi_{X_i}(r)$ of the projections might be a *strict superset* of the original relation instance r. In this case, in general an observer of the projections cannot decide whether a *specific* tuple generated by the join is spurious, i.e., not contained in the original relation instance. However, if the observer knows the original cardinality, he can easily decide whether or not the join has produced spurious tuples, just by comparing $||r||$ with $||\bowtie_{j=1\ldots m} \pi_{X_i}(r)\,||$. Further we remind that for pairwise disjoint attribute sets X_i the join $\bowtie$ degenerates to the Cartesian product $\times$.

Lemma 1 (Matching-inferrable binary value combination). *Let n be the number of (sub)tuple occurrences in a relation instance r and the fragment instances f_i and f_j, $i \neq j$, of a relational schema $(R(X), \mathcal{SC})$ with $\mathcal{SC} = \emptyset$ (i.e., without data dependencies) and the fragmentation schema $\mathcal{F} = \langle F_1(X_1), \ldots, F_m(X_m)\rangle$, respectively. Furthermore, let $C \subseteq X_i \cup X_j$ be an association syntactically protected by $\mathcal{F}$. Consider any value combination $\mu = (\mu_i, \mu_j)$ over C such that μ_l occurs in exactly c_{μ_l} many subtuple instances of f_l, for $l \in \{i, j\}$. Then, the following assertions hold:*

1. *If $c_{\mu_i} + c_{\mu_j} > n$, then μ is matching-inferrable and has at least $c_{\mu_i} + c_{\mu_j} - n$ occurrences in any matching M.*
2. *If $c_{\mu_i} + c_{\mu_j} \leq n$ and r is unique on $X \setminus (X_i \cup X_j)$, i.e., the projection of r on $X \setminus (X_i \cup X_j)$ would not produce duplicates, then μ is* not *matching-inferrable.*

Proof. 1. Assume $c_{\mu_i} + c_{\mu_j} > n$, and consider any matching M. For the c_{μ_i} many subtuple instances of f_i containing μ_i there are at most $n - c_{\mu_j}$ many subtuple instances in f_j that do *not* contain μ_j. Hence $c_{\mu_i} - (n - c_{\mu_j}) = c_{\mu_i} + c_{\mu_j} - n \geq 1$ many of the former subtuple instances must be matched with a subtuple instance of f_j containing μ_j.

2. We have to show that there exists an $\mathcal{SC}$-admissible matching M such that $M(r)$ is a set (without duplicates) not containing μ. If r itself does not contain μ, then the matching that exactly undoes the fragmentation has the desired properties. Otherwise, we can remove all occurrences of $\mu = (\mu_i, \mu_j)$ without affecting the fragmentation result: for each tuple with such an occurrence we exchange either the X_i- or the X_j-component with the respective component of a tuple that contains neither μ_i nor μ_j. Such a tuple exists by the first assumption that $c_{\mu_i} + c_{\mu_j} \leq n$. Let then M be a matching such that $M(r)$ generates the result of all exchanges. By the uniqueness of r on $X \setminus (X_i \cup X_j)$ according to the second assumption, $M(r)$ has no duplicates. □

Remark 4 (Impact of duplicate-free relation instances). Unfortunately, in Lemma 1 the condition $c_{\mu_i} + c_{\mu_j} > n$ is not necessary for μ being matching-inferrable, as witnessed by the following counterexample. Let $X = \{A_i, A_j\}$ and $r = \{\, (a, \mu_j), (\mu_i, \mu_j), (a, b) \,\}$ be a relation instance having $n = 3$ tuples with $a \neq \mu_i$ and $b \neq \mu_j$ and, thus $c_{\mu_i} + c_{\mu_j} = 1 + 2 = 3 = n$. Consider any matching M of the fragments $f_i = \{\!\{\, (a), (\mu_i), (a) \,\}\!\}$ and $f_j = \{\!\{\, (\mu_j), (\mu_j), (b) \,\}\!\}$ such that $M(r)$ does not contain $\mu = (\mu_i, \mu_j)$. Then M combines μ_i with b and, thus, both occurrences of μ_j with a, yielding duplicates. Hence, μ is matching-inferrable. In contrast, the proof of Lemma 1, assertion 2 shows that the condition $c_{\mu_i} + c_{\mu_j} > n$ would be necessary if we allowed duplicates in original relation instances.

Theorem 1 (Existence of a matching-inferrable value combination). *Let n be the number of (sub)tuple instances in a relation instance r and the fragment instances $f = \langle f_1, \ldots, f_m \rangle$ of a relational schema $(R(X), \mathcal{SC})$ with $\mathcal{SC} = \emptyset$ (i.e., without data dependencies) and the fragmentation schema $\mathcal{F} = \langle F_1(X_1), \ldots, F_m(X_m) \rangle$, respectively. Furthermore, let $C \subseteq X_{i_1} \cup \cdots \cup X_{i_k}$, with $P := \{\, i \mid C \cap X_i \neq \emptyset \,\} = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, m\}$, be an association syntactically protected by $\mathcal{F}$, and $maxc_{i_l}$ the maximal number of occurrences of a subtuple over the attributes of $C \cap X_{i_l}$ in f_{i_l}, for $l = i_1, \ldots, i_k$.*

If[2] $maxc_{i_1} + \cdots + maxc_{i_k} > (k-1) \cdot n$, then there exists a matching-inferrable value combination $\mu = (\mu_{i_1}, \ldots, \mu_{i_k})$ over $C \cap X_{i_1} \cup \cdots \cup C \cap X_{i_k}$.

Proof. For $k = 2$, the theorem is an immediate consequence of the first assertion of Lemma 1. The general case can be proved by induction on k, exploiting the induction hypothesis and again the first assertion of Lemma 1. □

[2] As discussed above, if we allowed duplicates in original relation instances, the condition would also be necessary.

4 Inferences with Numerical Dependencies

Even if we allowed duplicates in original relation instances, in general the condition presented in Theorem 1 would not be necessary for a relational schema $(R(X), \mathcal{SC})$ with nontrivial *data dependencies* in $\mathcal{SC}$ such that certain sets of tuple instances over X are not accepted as a relation instance of the schema. In fact, a data dependency might relate the parts of an occurrence of a value combination split among different fragments. Moreover, to exploit such a relationship for an inference attack sometimes the knowledge of the frequencies of potentially combined parts is crucial.

In this section, we restrict our investigations to cardinality constraints in the form of *numerical dependencies* which include *functional dependencies* as a special case. We will consider the class of *tuple-generating dependencies* in the next section. In this study, however, we neither intend to cover the full range of data dependencies considered so far nor to relate the chosen examples exactly to the various versions suggested in the literature. Rather, by means of examples seen to be intuitively representative, we aim to demonstrate the issues of unwanted and sometimes even unexpected inferences enabled by the knowledge of data dependencies. Regarding the comprehensive class of data dependencies we refer the reader to, e.g., the extensive surveys of the rich literature contained in the textbooks [1,27] and a few original contributions [3,25,4,21,5,16,26,22,8] selected out of many more works.

Example 2 (Functional dependency and frequencies). Consider the relation instance of the schema *Patient* and the fragmentation shown in Figure 1 and Figure 2, respectively. The relation instance satisfies the functional dependency MEDICATION $\rightarrow$ ILLNESS. So, we now assume that this semantic constraint has publicly been declared for the schema such that the attacking receiver holds only those relation instances possible that satisfy this functional dependency. The value "MedB" under attribute MEDICATION occurs twice in the fragment instance f_3, and thus also in the hidden relation instance. By the semantic constraint, in the hidden relation instance, both occurrences must appear in combination with the same value under attribute ILLNESS, which then must occur at least twice. Since duplicates are preserved, this value must also occur at least twice in the fragment instance f_2. Only the value "Laceration" meets this requirement. Thus, seeing the fragment instances and knowing the functional dependency enables to infer that the value combination (Laceration, MedB) over the attribute set $\{$ILLNESS, MEDICATION$\}$ occurs in the hidden relation instance, though, in the sense of Definition 2, this attribute set is an association syntactically protected by the fragmentation and the condition of Lemma 1, assertion 1 is not satisfied.

The preceding example is captured by the following proposition. For the sake of simplicity, it is expressed in terms of a functional dependency relating two single attributes A and B of some schema with attribute set X. Evidently, for the general case of a functional dependency relating two sets of attributes $Y \subseteq X$ and $Z \subseteq X$, a suitably adapted proposition holds as well.

Proposition 1 (Inferences by equations on frequencies). *For $A, B \in X$, let $(R(X), \{A \to B\})$ be a relational schema with the functional dependency $A \to B$ as a single semantic constraint and r a relation instance containing n different tuples. Furthermore, for each value a occurring in r under attribute A let c_a be the number of its occurrences and, similarly, for each value b occurring in r under attribute B let c_b be the number of its occurrences. Then, for all values b occurring in r under attribute B the following equation holds:*

$$\sum_{a \in \pi_A(\sigma_{B=b}(R))} c_a = c_b \, . \tag{1}$$

Proof. Consider any value b occurring in r under attribute B. Then $\pi_A(\sigma_{B=b}(R))$ is the set of all values a – without duplicates – such that (a, b) occurs in the relation instance r of R. Each such value a occurs c_a many times, and by the functional dependency $A \to B$ each occurrence is together with b. □

If an attacker knows both the fragment f_A showing the column A of r and the fragment f_B showing the column B of r, he can exploit the preceding proposition in a straightforward way as follows. Seeing both fragments, the attacker also knows all frequencies c_a and c_b. He can then simply attempt to solve the set of equations derived from instantiating the equation (1) – with the relation symbol R treated as the unknown item – to infer all possibilities for the hidden relation instance r. If there is a unique solution, the attacker has completely inferred the duplicate-preserving $\{A, B\}$-part of the hidden relation instance r from its published fragments f_A and f_B. Even otherwise, all solutions might still coincide for a particular value b whose combinations in the hidden relation instance r are then revealed.

We can consider the attacker's task as to solve the following variant of a *packing problem*. We interpret each value b occurring under attribute B in f_B as a container having capacity c_b, and each value a occurring under attribute A in f_A as a packet of size c_a. Then the attacker has to find all pairs (a, b) that appear in each possible allocation of the packets to the containers such that all containers are completely filled, under the preconditions that (1) the sum over the packet sizes equals the sum over the container capacities and (2) there exists a solution, namely the one induced by the original relation instance.

Our variant is closely related to the NP-complete problem [SR1] BIN PACKING described in [20], where k bins (containers) each of the same capacity B should be filled with a finite set of items (packets) of given sizes. In our variant the bins may have different capacities, the existence of a solution completely filling all bins is known beforehand by the precondition, and we are interested in the allocations common to all solutions. Moreover, another related problem from the field of protection of statistical databases, [SR35] CONSISTENCY OF DATABASE FREQUENCY TABLES, is listed in [20] as being NP-complete. For this problem, a frequency refers to the number of occurrences of a *pair* of values under *any two* different attributes; furthermore, knowledge about value combinations is also not restricted to the content of a fragment. The problem then is

F_A	A
	a_1
	a_1
	a_2
	a_3

F_B	B
	b_1
	b_1
	b_2
	b_3

F_A	A
	a_1
	a_1
	a_1
	a_1
	a_2
	a_2
	a_2
	a_3

F_B	B
	b_1
	b_1
	b_1
	b_1
	b_2
	b_2
	b_2
	b_2

F_A	A
	a_1
	a_1
	a_1
	a_1
	a_1
	a_2
	a_2
	a_2
	a_3

F_B	B
	b_1
	b_1
	b_1
	b_1
	b_1
	b_2
	b_2
	b_2
	b_2

Fig. 3. Three fragmentations (f_A^i, f_B^i) of different relation instances of a schema with functional dependency $A \rightarrow B$

to decide whether there exist unknown values supplementing the already known ones such that the given frequencies are satisfied. Due to these relationships, we expect that the attacker's task will be of high computational complexity. This expectation is also supported by the formal complexity analysis given in [7] and other works in the field of confidentiality-preserving data publishing, elaborated from the point of view of the defender.

Example 3 (Functional dependency and frequencies for packing problem). Given the functional dependency $A \rightarrow B$, consider the three fragmentation instances (f_A^i, f_B^i) over the same fragmentation schema $\langle F_A(\{A\}, F_B(\{B\}\rangle$ shown in Figure 3. Basically, (f_A^1, f_B^1) is an abstract version of Example 2: since "packet" a_1 can only be allocated to "container" b_1, the value combination (a_1, b_1) must occur twice in the original hidden relation instance underlying this fragmentation; nothing more definite can be inferred about the combinations of a_2 and a_3 with b_2 and b_3, respectively. For (f_A^2, f_B^2), "packet" a_1 can be allocated to either "container" b_1 or "container" b_2, and then "packets" a_2 and a_3 both must be allocated to the container not selected for "packet" a_1. Thus, no (definite) inferences are possible at all. For (f_A^3, f_B^3), Lemma 1 already asserts that the value combination (a_1, b_1) is matching-inferrable and hence occurs at least once in the hidden instance underlying this fragmentation, since $c_{a_1} + c_{b_1} = 5 + 5 > 9$; under the presence of the functional dependency, now Proposition 1 implies more, namely that there must be exactly 5 occurrences. In turn, the latter fact together with the observable frequencies imply that (a_2, b_2) occurs 3 times in the hidden instance, and (a_3, b_2) once. Hence, in this case, the duplicate-preserving $\{A, B\}$-part of the hidden instance can be completely reconstructed from the observable fragmentation instances.

Though even more complex, we can extend our considerations to *numerical dependencies* of the form $Y \rightarrow_{max}^{min} Z$, where Y and Z are sets of attributes and $1 \leq min \leq max$ are integers. Such a numerical dependency requires that each subtuple μ over the attributes of Y occurs combined with at least min and at most max different subtuples ν over the attributes of Z. A functional dependency $Y \rightarrow Z$ can be seen as a numerical dependency $Y \rightarrow_1^1 Z$.

R	A	B
	a_1	b_1
	a_1	b_2
	a_2	b_1
	a_2	b_2
	a_3	b_1
	a_3	b_2

F_A	A
	a_1
	a_1
	a_2
	a_2
	a_3
	a_3

F_B	B
	b_1
	b_1
	b_1
	b_2
	b_2
	b_2

Fig. 4. A relation instance of a schema with numerical dependency $A \rightarrow_2^2 B$ and (unscrambled) derived fragmentation instances that uniquely determine the set of tuples occurring in the relation instance

Example 4 (Numerical dependency and frequencies). Figure 4 illustrates a special case of an inference for a numerical dependency $A \rightarrow_{max}^{min} B$ with $k := min = max$ and a relation instance with $l \cdot k$ tuple occurrences for some number l, having l many different values under attribute A and k many different values under attribute B. By the dependency, each of the l values under A must occur in the hidden relation instance combined with each value under B, and thus the relation instance must be the Cartesian product of the value sets involved.

Proposition 2 (Inferences by equations on frequencies). *For $A, B \in X$, let $(R(X), \{A \rightarrow_{max}^{min} B\})$ be a relational schema with the numerical dependency $A \rightarrow_{max}^{min} B$ as a single semantic constraint and r a relation instance containing n tuples. For each value a occurring in r under attribute A, let c_a be its frequency, i.e., the number of its occurrences under A, and $d_a := \|\, \pi_B(\sigma_{A=a}(R)) \,\|$ its diversity, i.e., the number of different values under attribute B occurring together with a, and then for each $b \in \pi_B(\sigma_{A=a}(R))$, c_a^b the frequency of (a, b), i.e., the number of its occurrences in r. Moreover, for each value b occurring in r under attribute B let c_b be the number of its occurrences under B.*

Then, for all values a occurring in r under attribute A and for all values b occurring in r under attribute B the following equations holds:

$$c_b = \sum_{a \in \pi_A(\sigma_{B=b}(R))} c_a^b \,, \tag{2}$$

$$c_a = \sum_{b \in \pi_B(\sigma_{A=a}(R))} c_a^b \,, \tag{3}$$

$$min \leq d_a \leq max \,. \tag{4}$$

Proof. Equation (2) can similarly be justified as equation (1). Equations (3) and (4) are immediate consequences of the definitions of the items involved. □

Remark 5. Similarly as discussed for functional dependencies, the attacker's task can be considered as solving the set of equations derived from instantiating the equations (2), (3) and (4) – with the relation symbol R and the values d_a and c_a^b derived from R treated as the unknown items. Again, in general the equations will not have a unique solution and, thus, regarding a syntactically protected association C, conceptually only the intersection of the projections on C will deliver a certain inference. Surely, the data owner is interested in blocking such an inference, i.e., in ensuring that the intersection in empty. Evidently, the latter task is computationally highly complex, and so far neither a practical procedure to solve the equations involved nor an efficient and effective method to block their solvability is known to us. Presumably, the best we can hope to achieve is an approximative blocking method, favoring efficiency at the costs of loss of availability or violation of strict confidentiality.

5 Inferences with Tuple-Generating Dependencies

We are now addressing the impact of another well-known class of data dependencies, namely *tuple-generating dependencies* which, basically, require that whenever one or more tuples each of a specific form occur together in a relation instance, possibly related by identical components, another tuple partially constructed from selected components of those tuples and some constant symbols has to be present as well. Schema design theory has identified such dependencies as a possible source of redundancy in relation instances and, thus, of options to infer nontrivial information already from parts of an instance. Furthermore, each functional dependency entails a corresponding tuple-generating dependency, and thus studying the latter kind another aspect of the former one will be treated, together with the consequences of extensional background knowledge as expressed by means of constant symbols.

Definition 4 (Tuple-generating dependencies). *For a relational schema $(R(X), \mathcal{SC})$ with attribute set $X = \{A_1, \ldots, A_n\}$ and data dependencies $\mathcal{SC}$, an element of $\mathcal{SC}$ is called a* tuple-generating dependency *if it has a representation as an (untyped) sentence (without free variables) of first-order logic (with constant symbols) of the syntactic (implicational) form*

$$(\forall \boldsymbol{x})(\exists \boldsymbol{y})[\,[\bigwedge_{j=1,\ldots,p} \alpha_j] \implies \beta\,] \quad \textit{such that}$$

1. *for $j = 1, \ldots, p$, the* premises *α_j are relational atoms of the form $R(t_{j,1}, \ldots, t_{j,n})$ where each term $t_{j,i}$ is either a universally quantified variable contained in $\boldsymbol{x}$ or a constant symbol;*
2. *the* conclusion *β is a relational atom of the form $R(t_{p+1,1}, \ldots, t_{p+1,n})$ where each term $t_{p+1,i}$ is either a universally quantified variable contained in at least one premise (and thus also in $\boldsymbol{x}$) or an existentially quantified variable contained in $\boldsymbol{y}$ or a constant symbol;*
3. *the* prefix *$(\forall \boldsymbol{x})(\exists \boldsymbol{y})$ comprises exactly the variables occurring in some premise or in the conclusion.*

We start our investigations about the impact of a single tuple-generating dependency by considering two examples.

Example 5 (Tuple-generating dependency without frequencies). Consider the fragmentation schema with attribute sets $X_1 = \{A_1, A_3\}$ and $X_2 = \{A_2, A_4\}$ to split the syntactically protected association $C = \{A_3, A_4\}$ for the relational schema with the attributes A_1, A_2, A_3 and A_4 and the dependency Φ defined by

$$(\forall x_1, x_2, x_3, x_4, \bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4)$$
$$[\,[R(x_1, \bar{x}_2, x_3, \bar{x}_4) \wedge R(\bar{x}_1, x_2, \bar{x}_3, x_4)] \implies (\exists y_1, y_2)R(y_1, y_2, x_3, x_4)]\,.$$

Having the fragmentation schema in mind, this dependency can be given the following intuitive interpretation: if simultaneously a value combination (x_1, x_3) is visible in the fragment for X_1 and a value combination (x_2, x_4) is visible in the fragment for X_2, then the value combination (x_3, x_4) occurs in the split association C, the intended protection of which would thus be violated. In contrast, lacking the background knowledge that the original relation satisfies Φ, in general an observer could not distinguish whether the value x_3 seen in one fragment and the value x_4 seen in the other fragment actually occur together in a single tuple under the attributes in C or not. In fact, the relation $\{(a_1, \bar{a}_2, a_3, \bar{a}_4), (\bar{a}_1, a_2, \bar{a}_3, a_4)\}$ would generate the fragments $\{(a_1, a_3,), (\bar{a}_1, \bar{a}_3)\}$ and $\{(\bar{a}_2, \bar{a}_4), (a_2, a_4)\}$, leaving open which of the two possible matchings is the original one, in particular whether the value combinations (a_3, a_4) and $(\bar{a}_3, \bar{a}_4)$ or the value combinations $(a_3, \bar{a}_4)$ and $(\bar{a}_3, a_4)$ actually occur under the attributes in C.
We also note that the inspected tuple-generating dependency Φ can actually be considered as an *embedded multivalued dependency* for the projection on the attribute set $\{A_3, A_4\}$, shortly denoted by $\emptyset \twoheadrightarrow A_3|A_4$, requiring that this projection is the Cartesian product of the projection on $\{A_3\}$ with the projection on $\{A_4\}$, and obviously violated by the relation defined above.

Proposition 3 below will treat the kind of situation described in Example 5 more generally. The proposition will present three syntactic conditions regarding the occurrences of terms in a tuple-generating dependency to perform successful inferences about a split association C solely on observing data in the fragments. These conditions are outlined as follows. On the one hand and straightforwardly, (a) each non-constant term in the C-part of the conclusion has to be determined in at least one premise. On the other hand and somehow more sophistically, the constraints on relevant terms as expressed in the premises have to be restricted (b) regarding occurrences of one or more terms within a single premise and (c) regarding the occurrences of one term across two or more premises.

In terms of first-order logic, the latter two conditions would allow us to rewrite the tuple-generating dependency using a slightly more general syntactic form where each original premise has been transformed into a derived formula that gets a prefix of existentially quantified variables. Such a purely existential prefix then serves to express an effect that is equivalent to consider only the projection on the attributes of only one of the fragments. For the dependency Φ

considered above, the following rewriting would be suitable:

$$\begin{array}{l}(\forall x_1, x_2, x_3, x_4) \\ [\,[(\exists \bar{x}_2, \bar{x}_4) R(x_1, \bar{x}_2, x_3, \bar{x}_4) \wedge (\exists \bar{x}_1, \bar{x}_3) R(\bar{x}_1, x_2, \bar{x}_3, x_4)] \\ \quad \Longrightarrow (\exists y_1, y_2) R(y_1, y_2, x_3, x_4)]\,.\end{array}$$

To formally express and verify the intuition just outlined, we first need to precisely define the notions of an attribute being either *essential* or *isolated*.

Definition 5. *Let α_j be a premise of a tuple-generating dependency Φ over the attribute set $X = \{A_1, \dots, A_n\}$. Then the set E_j of* essential *attributes (for α_j) is defined as the smallest subset of X with the following properties:*

1. *If $t_{j,i}$ is a constant symbol in α_j, then $A_i \in E_j$.*
2. *If t_{j,i_1} is a universally quantified variable multiply occurring in α_j such that $t_{j,i_1} = t_{j,i_2}$ with $i_1 \neq i_2$, then both $A_{i_1} \in E_j$ and $A_{i_2} \in E_j$.*
3. *If $t_{j,i}$ is a universally quantified variable in α_j that also occurs in the C-part of the conclusion β of Φ, then $A_i \in E_j$.*

All remaining attributes are called isolated *in α_j, i.e., we define $I_j = X \setminus E_j$.*

For the simple dependency Φ considered in Example 5 above, only the third rule applies and thus we get $E_1 = \{A_3\}$ and $E_2 = \{A_4\}$.

Proposition 3 (Inferences by a single tuple-generating dependency only). *Let $\mathcal{F} = \langle F_1(X_1), \dots, F_m(X_m) \rangle$ be the fragmentation schema derived from a relational schema $(R(X), \mathcal{SC})$ with attribute set $X = \{A_1, \dots, A_n\}$ and a sequence $\mathcal{X}$ of attribute sets partitioning X such that $\mathcal{SC} = \{\Phi\}$ contains the tuple-generating dependency $\Phi = (\forall \boldsymbol{x})(\exists \boldsymbol{y})[\,[\bigwedge_{j=1,\dots,p} \alpha_j] \implies \beta\,]$ as a single semantic constraint. Furthermore, let attribute set C be an association syntactically protected by $\mathcal{F}$, and consider the following assertions:*

1. *(a) In the conclusion $\beta = R(t_{p+1,1}, \dots, t_{p+1,n})$, for each attribute $A_i \in C$ the term $t_{p+1,i}$ is a constant symbol or a universally quantified variable.*
   (b) For each premise α_j the set E_j of its essential attributes is fully contained in exactly one attribute set $X_{e(j)}$ of the partition $\mathcal{X}$.
   (c) If a universally quantified variable x occurs in two or more premises $\alpha_{j_1}, \alpha_{j_2}, \dots$, then all occurrences are within the pertinent sets of essential attributes $E_{j_1}, E_{j_2}, \dots$.
2. *For all relation instances r of $(R(X), \mathcal{SC})$ satisfying the premises $\sigma[\alpha_1], \dots, \sigma[\alpha_p]$ of Φ for some substitution σ of the variables in $\boldsymbol{x}$ the generated fragmentation instance f is* not inference-proof *regarding C, i.e., $\mathcal{F}^{-1,C}_{\mathcal{SC}}(f) \neq \emptyset$.*

Then assertion 1 implies assertion 2.

Proof. Assuming assertion 1, suppose the relation instance r satisfies both the sentence Φ, which has only universally quantified variables in the premises, and the substituted premises $\sigma[\alpha_1], \ldots, \sigma[\alpha_p]$ of Φ for a suitable substitution σ of the variables in $\boldsymbol{x}$. Then there exists a substitution τ of the variables in $\boldsymbol{y}$ such that r also satisfies the substituted conclusion $\tau[\sigma[\beta]] = \tau[\sigma[R(t_{p+1,1}, \ldots, t_{p+1,n})]\,]$. Thus the subtuple μ over C formed from this conclusion occurs in r. According to the assumed assertion 1.(a), the substitution τ for the existentially quantified variables is not relevant for μ and thus we actually have $\mu = (\sigma[t_{p+1,i}])_{A_i \in C}$. In the remainder of the proof we will verify that for the fragmentation $\mathcal{F}(r) = f = \langle f_1, \ldots, f_m \rangle$ we have $\mu \in \mathcal{F}_{SC}^{-1,C}(f)$ and thus $\mathcal{F}_{SC}^{-1,C}(f) \neq \emptyset$.

Let $\tilde{r}$ be any relation instance of the relational schema $(R(X), \{\Phi\})$ generating the same fragmentation, i.e., $\mathcal{F}(\tilde{r}) = f$.

For $j = 1, \ldots, p$ define μ_j to be the $X_{e(j)}$-part of $\sigma[\alpha_j]$, where $e(j)$ is determined by assumption 1.(b) assuring that μ_j assigns values to all essential attributes of the premise α_j. In particular, we have $\mu_j \in f_{e(j)}$.

Furthermore, for any isolated attribute A_i of α_j consider the term $t_{j,i}$. According to property 1 of Definition 5, $t_{j,i}$ is not a constant symbol and, thus, it is a universally quantified variable. Moreover, this variable has no further occurrences within that premise or any other premise or the C-part of the conclusion, according to property 2 of Definition 5, the assumed assertion 1.(c) and property 3 of Definition 5, respectively.

Let $\boldsymbol{x}_{iso}$ comprise all those universally quantified variables under isolated attributes, and $\boldsymbol{x}_{ess}$ the remaining ones occurring in the essential parts of the premises. Denoting the restriction of the substitution σ to $\boldsymbol{x}_{ess}$ by σ_{ess} and observing that $\mu_j \in f_{e(j)} = \bar{\pi}^{?}_{X_{e(j)}}(\tilde{r})$ (here $\bar{\pi}^{?}$ signifies a projection in the sense of Definition 1), we conclude that there exists a substitution σ_{iso} of the variables $\boldsymbol{x}_{iso}$ such that $\sigma_{iso}[\sigma_{ess}[\alpha_j]\,] \in \tilde{r}$. By the construction, these tuples comply with the premises of the dependency Φ. Applying Φ then implies that for some substitution $\overline{\tau}$ of the existentially quantified variables in $\boldsymbol{y}$ also $\overline{\tau}[\sigma_{iso}[\sigma_{ess}[\beta]\,]\,] \in \tilde{r}$. By property 3 of Definition 5 and assumed assertion 1.(a) the C-part of the tuple $\overline{\tau}[\sigma_{iso}[\sigma_{ess}[\beta]\,]\,]$ only depends on σ_{ess} and thus equals μ. Hence $\mu \in \pi_C(\tilde{r})$. □

Proposition 3 describes situations that enable an attacking observer to violate inference-proofness just be logical entailment without additionally exploiting frequencies. The next example tells us that even in situations not captured by Proposition 3 the observation of frequencies might turn out to be harmful.

Example 6 (Tuple-generating dependency and frequencies). We reconsider Example 5 above but now assume a more frequently encountered kind of a tuple-generating dependency, namely the *multivalued dependency* shortly denoted by $A_1, A_2 \twoheadrightarrow A_3|A_4$, having the following formalization in first-order logic:

$$(\forall x_1, x_2, x_3, x_4, \bar{x}_3, \bar{x}_4)$$
$$[\,[R(x_1, x_2, x_3, x_4) \wedge R(x_1, x_2, \bar{x}_3, \bar{x}_4)] \implies R(x_1, x_2, x_3, \bar{x}_4)]\,.$$

Intuitively, this dependency requires that whenever the value combination (x_1, x_2) under the attributes A_1 and A_2 occurs both with the value combina-

R	A_1	A_2	A_3	A_4
	a_1	a_2	a_3	a_4
	$\bar{a}_1$	$\bar{a}_2$	$\bar{a}_3$	$\bar{a}_4$
	$\bar{a}_1$	$\bar{a}_2$	$\bar{a}_3$	$\dot{a}_4$
	$\bar{a}_1$	$\bar{a}_2$	$\dot{a}_3$	$\bar{a}_4$
	$\bar{a}_1$	$\bar{a}_2$	$\dot{a}_3$	$\dot{a}_4$

F_{A_1,A_3}	A_1	A_3
	a_1	a_3
	$\bar{a}_1$	$\bar{a}_3$
	$\bar{a}_1$	$\bar{a}_3$
	$\bar{a}_1$	$\dot{a}_3$
	$\bar{a}_1$	$\dot{a}_3$

F_{A_2,A_4}	A_2	A_4
	a_2	a_4
	$\bar{a}_2$	$\bar{a}_4$
	$\bar{a}_2$	$\dot{a}_4$
	$\bar{a}_2$	$\bar{a}_4$
	$\bar{a}_2$	$\dot{a}_4$

Fig. 5. A relation instance of a schema with multivalued dependency $A_1, A_2 \twoheadrightarrow A_3|A_4$ and derived fragmentation instances for attribute sets $X_1 = \{A_1, A_3\}$ and $X_2 = \{A_2, A_4\}$ that uniquely determine the set of tuples occurring in the relation instance

tions (x_3, x_4) and $(\bar{x}_3, \bar{x}_4)$ under the attributes A_3 and A_4, then the former value combination also occurs together with $(x_3, \bar{x}_4)$. More generally, this requirement then implies that (x_1, x_2) even occurs together with each element in the Cartesian product of the jointly occurring values under A_3 and the jointly occurring values under A_4. Consequently, for each value combination (x_1, x_2) the number of jointly occurring value combinations (x_3, x_4) is the cardinality of a Cartesian product. More precisely, this number is the arithmetic product of the cardinality of the jointly occurring x_3-values and the cardinality of the jointly occurring x_4-values. This consequence might enable combinatorial reasoning under the additional provision that the effect of duplicates is appropriately considered, i.e., that in general an observer can directly determine frequencies rather than only cardinalities.
Most notably, such a reasoning might be successful (from the point of view of an attacker) even for the present situation where both the attribute set $\{A_1, A_2\}$ of the dependency's left-hand side and the attribute set $\{A_3, A_4\}$ of the dependency's right hand side – which is the syntactically protected association C – are split by the fragmentation schema.
In fact, Figure 5 shows such a success as follows. The single occurrence of (a_1, a_3) has to match the single occurrence of (a_2, a_4), since otherwise, to complete the matching, for each fragment there would be only three candidates left to come up with a result of the form $(\bar{a}_1, \bar{a}_2, .\,, .\,)$ but the multivalued dependency would require that there are four, a contradiction. Furthermore, each matching of the remaining subtuple instances in the two fragments produces the same four tuples. Thus the original relation instance can be fully reconstructed based on the fragment instances.

6 Related Work and Conclusions

Inference analysis and control for information published about database relations have been an important topic in research on confidentiality enforcement since quite a long time, presumably starting with seminal work on information leakage via statistical databases, as, e.g., summarized by D. Denning [17] as early as 1982, and later continued under a much broader perspective, as more recently surveyed by, e.g., B. Fung et al [18]. The particular proposal of fragmenting relational data for ensuring confidentiality has arisen in different forms with the trend of outsourcing data for cloud computing since around ten years. First proposals by V. Ciriani et al [12,13] only used fragmentation, but subsequent work of G. Aggarwal et al [2], V. Ciriani et al [14], V. Ganapathy et al [19] and X. Xu et al [28] additionally employed encryption.

Initial deeper analysis by J. Biskup et al [11,10] of the actual achievements of fragmentation in the presence of data dependencies – as usually employed for relational databases in practice – pointed to the weakness of the simple syntactic splitting approach. This analysis also led to more semantically oriented refinements, guaranteeing inference-proofness in a strong sense for special classes of data dependencies, but unfortunately in general also increasing the computational complexity of finding appropriate fragmentation schemas. For the specific setting of [14] also underlying our contribution, V. Ciriani et al [15] later also provided a refinement and its analysis, considering a non-standard and still weak syntactic notion of functional dependence as an attacking receiver's background knowledge. Their refinement exemplifies a compromise between the conflicting goals involved, effective preservation of confidentiality on the one hand and efficient computation of fragmentations on the other hand.

Our contribution presents an *exploratory study* regarding the former goal when using the setting of [14]. While the authors of [15] only deal with a weak notion of functional dependence between attributes on the schema level, we study the impact of classical, more expressive functional dependencies on the *instance level*, and we extend these investigations to further important classes of *data dependencies*. Moreover, considering the instance level, we identify the crucial role of *preserving duplicates* when fragmenting a relation instance: this feature opens the way for *combinatorial reasoning* to infer hidden information, as far as we are aware for this context neglected in previous work. Once opened, for the first time this way enables us to investigate possible *interferences* of combinatorial reasoning about observable frequencies on the one hand and entailment reasoning about data dependencies known from the database schema and actual data values observed in the fragments on the other hand.

Briefly summarized, these investigations provide initial, hopefully representative insight into conditions that enable or block inferences of information that is intended to be hidden by completely splitting the underlying data, respectively. Though already aiming at covering most of the relevant cases, future work still has to complement the overall picture, ideally in order to come up with a complete characterization of inference options in terms of a condition that is both necessary and sufficient. Based on such a characterization, we could then design a refined fragmentation approach that guarantees inference-proofness just by ensuring that the result does not satisfy that condition. This long-term research project could be elaborated both for single relation instances and, even more ambitiously, for database schemas dealing with all their respective relation instances. We have already explored a first step in such a direction in our study [10], which however deals with a kind of fragmentation that let each observer only see one fragment. In contrast, one peculiarity of our setting as adopted from [14] is that one observer sees all fragments and, thus, might exploit multiple views on the same underlying original data. This situation is well-known to often constitute a threat to confidentiality, as, e.g., treated in [18] for different settings.

From the point of view of database theory, our contribution deals with a specific case of the much more general problem of computing the *inversion* of database queries or, equivalently, *solving equations* in the relational algebra, as studied in [9]. However, we are now deviating from the pure relational model, which treats relations as pure sets allowing no duplicates and incorporating no sequence of their members. Though we assume that the original relation is duplicate-free, we explicitly study the impact of maintaining duplicates in the fragments. Moreover, the inversion of a fragmentation by means of exploring matchings in our sense aims at undoing the deliberate scrambling of data representations. These features also make our settings slightly different from those for the classical studies of *lossless joins*, see, e.g., [1]. While joins more generally deal with overlapping projections, we require attribute-disjointness of the fragments which reduces the join to the Cartesian product and in most practical cases violates losslessness. Nevertheless, for this special case our results provide insight about the detailed *information content* of a lossy operation, see. e.g., [23].

Furthermore, in general inversion generates *uncertainty* about which element of the pre-image has been the actual one and, thus, the challenge arise how to determine the *certain* part of the pre-image contained in all its elements, also known as *skeptical reasoning*, see, e.g., [24]. It would be worthwhile to explore how the rich insight already gathered for this field could be adapted to our problem, which exhibits the following similarities and particularities. First, the space of possibilities is defined by two arguments: in both cases by a set of explicitly expressed data dependencies – and their closure under entailment of course – on the one hand and by visible data either original but incomplete one or by fragmentation derived one, respectively, on the other hand. Second, the aim is either to exactly determine the certain part (which would be the interest of an attacker) or to block all options to gain any certain information, which is the basic task of an owner's protection mechanism.

Acknowledgment: We would like to thank Manh Linh Nguyen for stimulating discussions while he has prepared his master thesis on a partial analysis of the approach of fragmentation with encryption to protect privacy in data storage.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading, MA (1995)
2. Aggarwal, G., Bawa, M., Ganesan, P., Garcia-Molina, H., Kenthapadi, K., Motwani, R., Srivastava, U., Thomas, D., Xu, Y.: Two can keep a secret: A distributed architecture for secure database services. In: 2nd Biennial Conference on Innovative Data Systems Research, CIDR 2005. pp. 186–199. Online Proceedings (2005)

3. Armstrong, W.W.: Dependency structures of data base relationships. In: IFIP Congress. pp. 580–583 (1974)
4. Beeri, C., Vardi, M.Y.: Formal systems for tuple and equality generating dependencies. SIAM J. Comput. 13(1), 76–98 (1984), https://doi.org/10.1137/0213006
5. Benczúr, A., Kiss, A., Márkus, T.: On a general class of data dependencies in the relational model and its implication problem. Computers Math.Applic. 21(1), 1–11 (1991)
6. Biskup, J.: Selected results and related issues of confidentiality-preserving controlled interaction execution. In: Gyssens, M., Simari, G.R. (eds.) 9th International Symposium on Foundations of Information and Knowledge Systems, FoIKS 2016. Lecture Notes in Computer Science, vol. 9616, pp. 211–234. Springer (2016)
7. Biskup, J., Bonatti, P.A., Galdi, C., Sauro, L.: Optimality and complexity of inference-proof data filtering and CQE. In: Kutylowski, M., Vaidya, J. (eds.) 19th European Symposium on Research in Computer Security, ESORICS 2014, Part II. Lecture Notes in Computer Science, vol. 8713, pp. 165–181. Springer (2014)
8. Biskup, J., Link, S.: Appropriate inferences of data dependencies in relational databases. Ann. Math. Artif. Intell. 63(3-4), 213–255 (2011), https://doi.org/10.1007/s10472-012-9275-0
9. Biskup, J., Paredaens, J., Schwentick, T., Van den Bussche, J.: Solving equations in the relational algebra. SIAM J. Comput. 33(5), 1052–1066 (2004), https://doi.org/10.1137/S0097539701390859
10. Biskup, J., Preuß, M.: Database fragmentation with encryption: Under which semantic constraints and a priori knowledge can two keep a secret? In: Wang, L., Shafiq, B. (eds.) Data and Applications Security and Privacy XXVII, DBSec 2013. Lecture Notes in Computer Science, vol. 7964, pp. 17–32. Springer (2013)
11. Biskup, J., Preuß, M., Wiese, L.: On the inference-proofness of database fragmentation satisfying confidentiality constraints. In: Lai, X., Zhou, J., Li, H. (eds.) Information Security, ISC 2011. Lecture Notes in Computer Science, vol. 7001, pp. 246–261. Springer (2011)
12. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Enforcing confidentiality constraints on sensitive databases with lightweight trusted clients. In: Data and Applications Security XXIII, DBSec 2009. Lecture Notes in Computer Science, vol. 5645, pp. 225–239. Springer (2009)
13. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Keep a few: Outsourcing data while maintaining confidentiality. In: 14th European Symposium on Research in Computer Security, ESORICS 2009. Lecture Notes in Computer Science, vol. 5789, pp. 440–455. Springer (2009)
14. 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 Transactions on Information and System Security 13(3), 22:1–22:33 (2010), Article no. 22
15. De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Livraga, G., Paraboschi, S., Samarati, P.: Fragmentation in presence of data dependencies. IEEE Trans. Dependable Sec. Comput. 11(6), 510–523 (2014), https://doi.org/10.1109/TDSC.2013.2295798
16. Demetrovics, J., Katona, G.O.H., Sali, A.: The characterization of branching dependencies. Discrete Applied Mathematics 40(2), 139–153 (1992), https://doi.org/10.1016/0166-218X(92)90027-8
17. Denning, D.E.: Cryptography and Data Security. Addison-Wesley, Reading, MA (1982)

18. Fung, B.C.M., Wang, K., Fu, A.W.C., Yu, P.S.: Introduction to Privacy-Preserving Data Publishing – Concepts and Techniques. Chapman & Hall/CRC, Boca Raton, FL (2011)
19. Ganapathy, V., Thomas, D., Feder, T., Garcia-Molina, H., Motwani, R.: Distributing data for secure database services. Transactions on Data Privacy 5(1), 253–272 (2012)
20. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York, NY (1979)
21. Grant, J., Minker, J.: Inferences for numerical dependencies. Theor. Comput. Sci. 41, 271–287 (1985), `https://doi.org/10.1016/0304-3975(85)90075-1`
22. Hartmann, S.: On the implication problem for cardinality constraints and functional dependencies. Ann. Math. Artif. Intell. 33(2-4), 253–307 (2001), `https://doi.org/10.1023/A:1013133428451`
23. Kolahi, S., Libkin, L.: An information-theoretic analysis of worst-case redundancy in database design. ACM Trans. Database Syst. 35(1), 5:1–5:32 (2010), `http://doi.acm.org/10.1145/1670243.1670248`
24. Libkin, L.: Certain answers as objects and knowledge. Artif. Intell. 232, 1–19 (2016), `https://doi.org/10.1016/j.artint.2015.11.004`
25. Sagiv, Y., Delobel, C., Parker Jr., D.S., Fagin, R.: An equivalence between relational database dependencies and a fragment of propositional logic. J. ACM 28(3), 435–453 (1981), `http://doi.acm.org/10.1145/322261.322263`
26. Sali Sr., A., Sali, A.: Generalized dependencies in relational databases. Acta Cybern. 13(4), 431–438 (1998)
27. Thalheim, B.: Entity-Relationship Modeling – Foundations of Database Technology. Springer (2000)
28. Xu, X., Xiong, L., Liu, J.: Database fragmentation with confidentiality constraints: A graph search approach. In: Park, J., Squicciarini, A.C. (eds.) 5th ACM Conference on Data and Application Security and Privacy, CODASPY 2015. pp. 263–270. ACM (2015)