Inference-Proof Data Publishing by Minimally Weakening a Database Instance*

Joachim Biskup and Marcel Preuß

Technische Universität Dortmund, Dortmund, Germany
{biskup,preuss}@ls6.cs.tu-dortmund.de

Abstract. Publishing of data is usually only permitted when complying with a confidentiality policy. To this end, this work proposes an approach to weaken an original database instance: within a logic-oriented modeling definite knowledge is replaced by disjunctive knowledge to introduce uncertainty about confidential information. This provably disables an adversary to infer this confidential information, even if he employs his a priori knowledge and his knowledge about the protection mechanism. As evaluated based on a prototype implementation, this approach can be made highly efficient. If a heuristic – resulting only in a slight loss of availability – is employed, it can be even used in interactive scenarios.

Keywords: A Priori Knowledge, Confidentiality Policy, Data Publishing, Disjunctive Knowledge, First-Order Logic, Inference-Proofness, Information Dissemination Control, k-Anonymity, Weakening.

1 Introduction

Nowadays, data publishing is ubiquitous. Governments are often legally obliged to provide data about matters of public concern, companies release project-related data to partners and even in most peoples' private lifes the sharing of data plays a major role. But usually only certain portions of some data are appropriate for being shared, as data often contains sensitive information. This applies in particular to data containing personal information, as surveyed in [11,23].

In the area of relational databases the logic-oriented framework of Controlled Interaction Execution (CIE) can assist a database owner in ensuring that each of his interaction partners can only obtain a so-called "inference-proof view" on the owner's data [3]. An inference-proof view does not contain information to be kept confidential from the respective partner, even if this partner is an adversary trying to deduce confidential information by drawing inferences based on his a priori knowledge and his general awareness of the protection mechanism.

An example of such a protection mechanism creating inference-proof materialized views – which are suitable for data publishing – by modifying a minimum number of truth-values of database tuples has been developed in [7]. This approach is rather versatile as it is based on an expressive fragment of first-order

* This work has been supported by the DFG under grant SFB 876/A5.

logic, but also suffers from this expressiveness because of its high computational complexity. Moreover, there might also be some ethical concerns as the modification of truth-values means that a user's view on the database contains lies.

This work introduces a novel approach within the framework of CIE creating *inference-proof materialized views* suitable for data publishing and thereby *provably* enforcing a confidentiality policy without modifying any truth-values: instead, harmful database tuples are replaced by weaker knowledge in the form of disjunctions formed by ground atoms stemming from the policy (each of which logically represents a database tuple). These disjunctions contain only true information, but weaken an adversary's possible gain in information such that the adversary is provably not able to infer protected sensitive information.

This approach is first developed in a *purely generic* way in the sense that non-trivial disjunctions of any length ≥ 2 might be employed. Then a possible instantiation of this generic approach is presented, which aims at *maximizing availability* in the sense that only disjunctions of length 2 are seen to be admissible. For this instantiation an algorithmic treatment based on graph clustering is given, which fully specifies the approach except for an admissibility criterion expressing which subsets of potential secrets might possibly form a disjunction. This criterion should be tailored to the needs of each specific application and can be easily specified by employing query languages of relational databases.

To be able to fully implement the availability-maximizing flavor to experimentally demonstrate its high efficiency – which can be even raised by employing a heuristic resulting only in a slight loss of availability – an example for such an admissibility criterion called *interchangeability* is provided and evaluated. Interchangeability admits only disjunctions formed by ground atoms which all pairwise differ in the same single position and do not differ in any other position. This local restriction of distortion preserves definite information about all but one position of each ground atom and *generalizes* each distorted value to a wider set of possible values. Moreover, extensions of the generic approach dealing with policies (and hence disjunctions) of existentially quantified atoms and also coping with a basic kind of an adversary's a priori knowledge are outlined.

As an adversary is aware of which values are weakened by simply considering the disjunctions, particular attention must be paid to eliminate so-called *meta-inferences* (cf. [3,5]). A deduction of sensitive information is called a meta-inference, if it is obtained by excluding all possible alternative settings, under which this sensitive information is *not* valid, by simulating these alternative settings as inputs for the algorithm generating the inference-proof views and by being able to distinguish the outputs resulting from each alternative setting from the published one. In this work meta-inferences are eliminated by imposing a total order on the sentences of weakened instances.

The generalization of values to a wider set of possible values is similarly used in the approaches of k-anonymization and ℓ-diversification [10,16,21]. These approaches aim at preventing the re-identification of individuals based on so-called quasi-identifiers, which describe some of the individuals' properties, by generalizing these quasi-identifiers. We could model k-anonymization and ℓ-diversification

as a special case within an extensions of our work, which deals with confidentiality policies containing disjunctions of existentially quantified atoms.

As the suppression of a value corresponds to its maximum generalization, this work is also related to the approach developed in [2], which aims at achieving confidentiality by replacing certain values of certain database tuples by null-values. But – in contrast to our work – this approach relies on the assumption that an adversary is *not* aware of which values are perturbed.

Moreover, there are other approaches clustering the vertices of a graph into sets of vertices to be made indistinguishable to achieve privacy [9,12]. But these approaches aim at preventing structural re-identification of the graph itself, while the approach presented in our work aims at achieving indistinguishability based on disjunctions induced from the clustering of the vertices of a graph.

In the remainder of this article, Sect. 2 provides the basic ideas of achieving inference-proofness by weakening a database instance. Sect. 3 then extends these ideas to also work with confidentiality policies of an arbitrary number of ground atoms, thereby balancing availability and confidentiality requirements. Subsequently, an overall algorithm – which is formally proved to comply with a declarative definition of inference-proofness – is presented in Sect. 4 and a prototype implementation of this algorithm is evaluated with respect to its efficiency in Sect. 5. Before concluding this work with Sect. 7, the algorithm is again extended in Sect. 6 to also deal with confidentiality policies containing existentially quantified atoms and to moreover consider an adversary's a priori knowledge.

2 Basic Ideas: Inference-Proofness by Weakening

The approach developed in this work is located within the area of relational databases. For simplicity, all data is supposed to be represented within a single database instance r over a relational schema $\langle R|\mathcal{A}_R|SC_R\rangle$ with relational symbol R and the set $\mathcal{A}_R = \{A_1, \ldots, A_n\}$ of attributes. Furthermore, all attributes are assumed to have the same *fixed but infinite* domain *Dom* of constants (cf. [4,15]) and the set SC_R contains some semantic (database) constraints (cf. [1]), which must be satisfied by the relational instance r. For now, these semantic constraints are neglected (i.e., $SC_R = \emptyset$), but they will become of interest in Sect. 6.

Each considered (original) instance r is supposed to represent *complete information*. Thus, the instance contains only a finite set of valid tuples and each constant combination $\boldsymbol{c}$ of the infinite set Dom^n with $\boldsymbol{c} \notin r$ is assumed to be *not* valid by Closed World Assumption (CWA). This is exemplified in Fig. 1(a).

In compliance with CIE (cf. [3,4,7,6,5]), a database instance is modeled logic-orientedly. Therefore, a language $\mathscr{L}$ of first-order logic containing the predicate symbol R of arity $|\mathcal{A}_R| = n$ and the binary predicate symbol $\equiv$ for expressing equality is set up. The fixed but infinite domain *Dom* is taken as the set of constant symbols of $\mathscr{L}$ and the variables of an infinite set *Var* can be used to build sentences (i.e., closed formulas) in the natural fashion [15].

This syntactic specification is complemented with a semantics reflecting the characteristics of databases by means of so-called DB-Interpretations [4,7,15]:

r	$+$	$-$
	(a,b,c)	(a,a,a)
	(a,c,c)	(a,a,b)
	(b,a,c)	(a,a,c)
		$\vdots$

(a) Complete instance r

$$R(a,b,c),\ R(a,c,c),\ R(b,a,c)$$
$$(\forall X)(\forall Y)(\forall Z)\ [$$
$$(X \equiv a\ \wedge\ Y \equiv b\ \wedge\ Z \equiv c)\ \vee$$
$$(X \equiv a\ \wedge\ Y \equiv c\ \wedge\ Z \equiv c)\ \vee$$
$$(X \equiv b\ \wedge\ Y \equiv a\ \wedge\ Z \equiv c)\ \vee$$
$$\neg R(X,Y,Z)\ ]$$

(b) Logic-oriented modeling of r

Fig. 1: Example of a logic-oriented modeling of a complete database instance

Definition 1 (DB-Interpretation). *Given the language $\mathscr{L}$ with the set Dom of constant symbols, an interpretation $\mathcal{I}$ is a* DB-Interpretation *for $\mathscr{L}$ iff*

(i) Dom is the universe of $\mathcal{I}$ and $\mathcal{I}(v) = v$ holds for each $v \in Dom$,
(ii) predicate symbol R is interpreted by a finite relation $\mathcal{I}(R) \subset Dom^n$,
(iii) predicate symbol $\equiv$ is interpreted by $\mathcal{I}(\equiv) = \{(v,v) \mid v \in Dom\}$.

A DB-Interpretation $\mathcal{I}_r$ is induced *by a complete database instance r, if its relation $\mathcal{I}_r(R)$ is instantiated by r, i.e., $\mathcal{I}_r(R) = \{\boldsymbol{c} \in Dom^n \mid \boldsymbol{c} \in r\}$.*

The notion of *satisfaction/validity* of formulas in $\mathscr{L}$ by a DB-Interpretation is the same as in usual first-order logic. A set $\mathcal{S} \subseteq \mathscr{L}$ of sentences *implies/entails* a sentence $\Phi \in \mathscr{L}$ (written as $\mathcal{S} \models_{DB} \Phi$) iff each DB-Interpretation $\mathcal{I}$ satisfying $\mathcal{S}$ (written as $\mathcal{I} \models_M \mathcal{S}$) also satisfies Φ (written as $\mathcal{I} \models_M \Phi$).

A logic-oriented modeling of the complete instance r of Fig. 1(a) is given in Fig. 1(b). Each valid tuple $\boldsymbol{c} \in r$ is modeled as a ground atom $R(\boldsymbol{c})$ of $\mathscr{L}$ and the infinite set of invalid tuples – which is not explicitly enumerable – is expressed implicitly by a so-called completeness sentence (cf. [4]) having a universally quantified variable X_j for each attribute $A_j \in \mathcal{A}_R$. This completeness sentence expresses that every constant combination $(c_1, \ldots, c_n) \in Dom^n$ (substituting the universally quantified variables $X_1, \ldots, X_n$) is either explicitly excluded from being invalid or satisfies the sentence $\neg R(c_1, \ldots, c_n)$. By construction, this completeness sentence is satisfied by the DB-Interpretation $\mathcal{I}_r$ induced by r.

To achieve confidentiality, a confidentiality policy containing so-called potential secrets [3] is set up. This policy is supposed to be known by an adversary trying to recover an original instance r unknown to him based on his knowledge about a weakened variant of r and his further (a priori) knowledge.

Definition 2 (Confidentiality Policy). *A* potential secret *Ψ is a sentence of $\mathscr{L}$ and a* confidentiality policy *psec is a finite set of potential secrets. A complete database instance r* obeys *a potential secret $\Psi \in psec$, if $\mathcal{I}_r \not\models_M \Psi$. Moreover, this instance r* obeys *the confidentiality policy psec, if r obeys each $\Psi \in psec$.*

For now – until Sect. 6 – only potential secrets in the form of ground atoms are considered. To enforce a given confidentiality policy *psec*, an incomplete weakened variant $weak(r, psec)$ of a complete original instance r over $\langle R|\mathcal{A}_R|\,\emptyset\,\rangle$ is constructed by a weakening algorithm such that

$$
\begin{array}{l}
R(b,a,c) \\
R(a,b,c) \vee R(a,c,c) \\
(\forall X)(\forall Y)(\forall Z)\ [\\
(X \equiv a \ \wedge\ Y \equiv b \ \wedge\ Z \equiv c) \ \vee \\
(X \equiv a \ \wedge\ Y \equiv c \ \wedge\ Z \equiv c) \ \vee \\
(X \equiv b \ \wedge\ Y \equiv a \ \wedge\ Z \equiv c) \ \vee \\
\neg R(X,Y,Z) \qquad\qquad\qquad]
\end{array}
$$

(a) Weakening $\mathit{weak}(r, \mathit{psec})$ obeying the policy $\mathit{psec} = \{R(a,\underline{b},c), R(a,\underline{c},c)\}$

$$
\begin{array}{l}
R(a,c,c),\ R(b,a,c) \\
R(a,b,c) \vee R(a,b,d) \\
(\forall X)(\forall Y)(\forall Z)\ [\\
(X \equiv a \ \wedge\ Y \equiv b \ \wedge\ Z \equiv c) \ \vee \\
(X \equiv a \ \wedge\ Y \equiv b \ \wedge\ Z \equiv d) \ \vee \\
(X \equiv a \ \wedge\ Y \equiv c \ \wedge\ Z \equiv c) \ \vee \\
(X \equiv b \ \wedge\ Y \equiv a \ \wedge\ Z \equiv c) \ \vee \\
\neg R(X,Y,Z) \qquad\qquad\qquad]
\end{array}
$$

(b) Weakening $\mathit{weak}(r, \mathit{psec}')$ obeying the policy $\mathit{psec}' = \{R(a,b,\underline{c}), R(a,b,\underline{d})\}$

Fig. 2: Possible inference-proof weakenings of the example instance of Fig. 1

- $\mathit{weak}(r, \mathit{psec})$ contains only true information, i.e., $\mathcal{I}_r \models_M \mathit{weak}(r, \mathit{psec})$, and
- for each potential secret $\Psi \in \mathit{psec}$ the existence of a complete alternative instance r^Ψ over $\langle R|\mathcal{A}_R|\ \emptyset\ \rangle$ is guaranteed such that
- this instance r^Ψ obeys Ψ, i.e., $\mathcal{I}_{r^\Psi} \not\models_M \Psi$, and the weakening of r^Ψ is indistinguishable from the weakening of r, i.e., $\mathit{weak}(r^\Psi, \mathit{psec}) = \mathit{weak}(r, \mathit{psec})$.

Given an original instance r and a *simple* policy $\mathit{psec} = \{\Psi_1, \Psi_2\}$, such a weakening $\mathit{weak}(r, \mathit{psec})$ can be easily computed: provided that Ψ_1 is *not* obeyed by r or (and, respectively) Ψ_2 is *not* obeyed by r, each knowledge about the constant combinations of Ψ_1 and Ψ_2 is removed from instance r and replaced by the weaker *disjunctive knowledge* that Ψ_1 or Ψ_2 is valid.

In contrast to the original instance r, a total order is supposed to be defined on the sentences that might occur in a weakened instance $\mathit{weak}(r, \mathit{psec})$ (cf. [4]). This guarantees that an alternative instance r^Ψ with $\mathcal{I}_{r^\Psi} \models_M \mathit{weak}(r, \mathit{psec})$ is *not* distinguishable from r based on a different arrangement of the sentences of its weakened instance $\mathit{weak}(r^\Psi, \mathit{psec})$ compared to $\mathit{weak}(r, \mathit{psec})$. Otherwise, an adversary might be able to draw the meta-inference (cf. Sect. 1) that r^Ψ is *not* the original instance of his interest because of $\mathit{weak}(r^\Psi, \mathit{psec}) \neq \mathit{weak}(r, \mathit{psec})$.

To exemplify the simple case, consider the potential secrets $\Psi_1 = R(a,b,c)$ and $\Psi_2 = R(a,c,c)$ both *not* obeyed by instance r of Fig. 1. Both Ψ_1 and Ψ_2 can be protected by weakening r as depicted in Fig. 2(a). From an adversary's point of view both alternative instances $r^{(1)} = \{(a,c,c),(b,a,c)\}$ obeying Ψ_1 and $r^{(2)} = \{(a,b,c),(b,a,c)\}$ obeying Ψ_2 are indistinguishable from the "real" original instance because of $\mathit{weak}(r, \mathit{psec}) = \mathit{weak}(r^{(1)}, \mathit{psec}) = \mathit{weak}(r^{(2)}, \mathit{psec})$.

Similarly, the potential secrets $\Psi_1' = R(a,b,c)$ *not* obeyed by r and $\Psi_2' = R(a,b,d)$ obeyed by r can be protected by weakening r as depicted in Fig. 2(b). In this case the completeness sentence known from Fig. 1(b) is extended by the disjunct $(X \equiv a \wedge Y \equiv b \wedge Z \equiv d)$ to ensure $\mathcal{I}_{r^{(1)'}} \models_M \mathit{weak}(r, \mathit{psec}')$ for the alternative instance $r^{(1)'} = \{(a,b,d),(b,a,c)\}$ obeying Ψ_1' as the constant combination (a,b,d) is not excluded from being invalid in r. The alternative instance obeying Ψ_2' is simply r itself.

As a last and easy case, consider a confidentiality policy $psec'' = \{\Psi_1'', \Psi_2''\}$ obeyed by r. Here no weakening of r is required, i.e., $weak(r, psec'') = r$.

3 Treating Non-Simple Sets of Potential Secrets

In Sect. 2 the basic ideas to create inference-proof weakenings protecting simple confidentiality policies have been introduced. Now, these basic ideas are extended to be able to deal with *non-simple* policies containing an arbitrary number of ground atoms. So, given a non-simple policy *psec*, the challenge is to construct a set of disjunctions consisting of potential secrets of *psec* such that availability and confidentiality requirements are suitably balanced.

3.1 A First Generic Approach

A first *generic approach* is to partition the policy *psec* into disjoint subsets called *clusters*. Then, for each cluster C a disjunction $\bigvee_{\Psi \in C} \Psi$ is constructed, provided that at least one potential secret of C is *not* obeyed by the original instance.

Note that a disjunction of length k is satisfied by $2^k - 1$ DB-Interpretations. Consequently, if $\mathcal{C}$ is the set of clusters, there are up to $\prod_{C \in \mathcal{C}} (2^{|C|} - 1)$ different (alternative) database instances, whose induced DB-Interpretations satisfy the weakened instance. From the point of view of an adversary only knowing the weakened instance, each of these instances is indistinguishable from the original one. Therefore, in terms of confidentiality it is desirable to construct large clusters to maximize the number of these instances, while in terms of availability small clusters are favored to minimize the number of these instances.

To also achieve a meaningful clustering of a policy *psec* regarding a specific application, an additional notion of *admissible indistinguishabilities* specifying all *admissible clusters* – i.e., all acceptable possibilities of making potential secrets of *psec* indistinguishable by disjunctions – should be provided. These admissible clusters need *not* be pairwise disjoint: the construction of a disjoint clustering $\mathcal{C}$, each of whose clusters is admissible, is the task of a *clustering algorithm*.

In some cases it might not be possible to construct such a clustering $\mathcal{C}$ and to moreover guarantee that each of its clusters has a certain minimum size k^*. As clusters of a suitable minimum size are inevitable to guarantee a wanted degree of confidentiality, one obvious approach is to *extend* each too small cluster $C \in \mathcal{C}$ of size $|C| < k^*$ by $k^* - |C|$ additional (i.e., artificial) potential secrets, thereby constructing an extended clustering $\mathcal{C}^*$ based on $\mathcal{C}$. But as each additional potential secret Ψ^A reduces availability, the goal is to find a clustering $\mathcal{C}$ for whose extension only a minimum number of additional potential secrets is needed.

For the quality of a weakening it is of crucial importance that the employed notion of admissible indistinguishabilities fits to the specific application considered: in terms of confidentiality all alternatives provided by a disjunction should be equally probable from an adversary's point of view and in terms of availability each disjunction should still provide as much useful information as possible.

Obviously, no generally valid approach to find such a notion for each possible specific application can be given. But as it is not desirable – and for policies of realistic size usually even impossible – to let a security officer manually design sets of admissible disjunctions, a generic method to construct admissible disjunctions based on a high level specification language is needed.

As a confidentiality policy should be usually managed by a database system, one possible approach to construct admissible clusters of size k (with $k \geq k^*$) is to compute a series of $k-1$ self-joins on the policy table – resulting in combinations each of which contains k pairwise different potential secrets. In this case well-known query languages such as SQL or relational algebra [19] let a security officer implement his concrete notion of admissible indistinguishabilities with the help of a corresponding join condition.

To allow the extension of too small clusters of size $k < k^*$, a notion of admissible indistinguishabilities might require that for some potential secrets of a confidentiality policy up to k^*-1 additional potential secrets can be constructed with a deterministic (and preferably efficient) algorithm. Such a construction only uses a finite subset of the domain *Dom* of constant symbols. So although favoring an *infinite* domain in theory (cf. Sect. 2) to avoid combinatorial effects possibly leading to harmful inferences, a "sufficiently large" *finite* domain is adequate in practice.

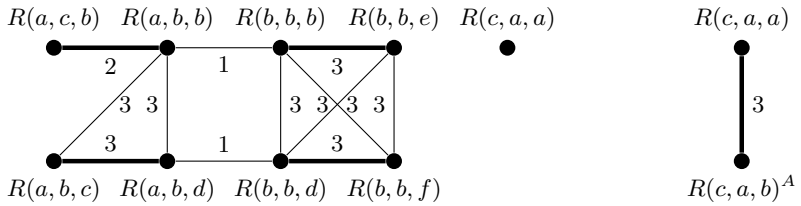
Definition 3 (Well-Defined Indistinguishability). *Given a confidentiality policy psec, the domain Dom of $\mathscr{L}$ and a minimum size k^* of clusters, a notion of admissible indistinguishabilities is* well-defined*, if there is a set $\mathcal{C}^*$ such that*

(i) for each $\Psi \in psec$ the set $\mathcal{C}^*$ contains a cluster $C_\Psi = \{\Psi, \Psi_{I_1}, \ldots, \Psi_{I_{k^*-1}}\}$ (possibly extended) such that $\Psi \neq \Psi_{I_i}$ for $1 \leq i \leq k^* - 1$ and $\Psi_{I_i} \neq \Psi_{I_j}$ for $1 \leq i < j \leq k^* - 1$ and $\bigvee_{\bar{\Psi} \in C_\Psi} \bar{\Psi}$ is an admissible indistinguishability,
(ii) $C_\Psi \cap C_{\Psi'} = \emptyset$ holds for all clusters $C_\Psi, C_{\Psi'} \in \mathcal{C}^*$ with $C_\Psi \neq C_{\Psi'}$,
(iii) there is a deterministic algorithm creating each (additional) Ψ^A of $\mathcal{C}^*$ with $\Psi^A \notin psec$, thereby (finitely) augmenting the active domain of psec and
(iv) the active domain of $\mathcal{C}^*$ is contained in Dom.

Note that an extension of clusters is generated independently of any database instance. As an adversary is moreover supposed to know the confidentiality policy as well as the deterministic algorithms employed, he is able to determine all additional potential secrets himself by simulating the corresponding algorithms.

3.2 Algorithmic Treatment of an Availability-Maximizing Flavor

In this subsection a possible instantiation of the generic approach aiming at *maximizing availability* – and hence keeping the size of clusters as small as possible – is developed. To be able to enforce confidentiality, for each potential secret Ψ the existence of at least one alternative instance obeying Ψ must be ensured (cf. Sect. 2 and Theorem 1 below). As clusters of size 2 – which are the smallest clusters complying with this requirement – correspond to binary relations, all admissible indistinguishabilities can be represented by a so-called indistinguishability-graph, whose edges represent all admissible clusters.



(a) Indistinguishability-graph with (bold) matching edges (b) Matching extension

Fig. 3: Graph with a clustering of potential secrets and a matching extension

Definition 4 (Indistinguishability-Graph). *Given a confidentiality policy psec and a well-defined notion of admissible indistinguishabilities, an* indistinguishability-graph *is an undirected graph $G = (V, E)$ such that*

(i) $V := psec$ *is the set of vertices of* G, *and the set of edges of* G *is*
(ii) $E := \{\, \{\Psi_1, \Psi_2\} \in V \times V \mid \Psi_1 \vee \Psi_2$ *is an admissible indistinguishability* $\}$.

An example of an indistinguishability-graph for a non-simple policy is given in Fig. 3(a). Note that for now the edge labelings are not of importance. On such a graph a maximum set of pairwise (vertex-)disjoint clusters of size 2 (i.e., edges) can then be computed efficiently with well-known maximum matching algorithms for general (i.e., not necessarily bipartite) graphs [14,17].

Definition 5 (Maximum Matching). *Let $G = (V, E)$ be an undirected graph (without loops). A subset $M \subseteq E$ is a* matching *on G, if $\{\Psi_1, \Psi_2\} \cap \{\bar{\Psi}_1, \bar{\Psi}_2\} = \emptyset$ for each pair of different* matching edges *$\{\Psi_1, \Psi_2\}, \{\bar{\Psi}_1, \bar{\Psi}_2\} \in M$. A matching M on G is a* maximum matching, *if $|M'| \leq |M|$ for each matching $M' \subseteq E$ on G. A maximum matching M on G is a* perfect matching, *if each vertex $\Psi \in V$ is* covered *by M, i.e., there is exactly one $\{\Psi_1, \Psi_2\} \in M$ with $\Psi \in \{\Psi_1, \Psi_2\}$.*

In Fig. 3(a) the subset of bold edges constitutes a maximum matching. As demonstrated, a maximum matching is not necessarily a perfect matching. Even given a connected graph with an even number of vertices, several vertices might remain uncovered by a maximum matching. To ensure that each potential secret is assigned to a cluster of size 2, additional potential secrets are created.

Definition 6 (Matching Extension). *Let psec be a confidentiality policy and let M be a maximum matching on the indistinguishability-graph of psec. A* matching extension *M^* of M and psec initially contains each matching edge $\{\Psi_1, \Psi_2\} \in M$ and subsequently, one after another, for each $\Psi \in psec$ uncovered by M an edge $\{\Psi, \Psi^A\}$ is added to M^*. Thereby Ψ^A is an* additional potential secret, *i.e., a deterministically created sentence $\Psi^A \notin psec$ of $\mathscr{L}$ such that $\Psi \vee \Psi^A$ is an admissible indistinguishability and $\Psi^A \notin \{\Psi_1, \Psi_2\}$ for each $\{\Psi_1, \Psi_2\} \in M^*$.*

In Fig. 3(b) such a matching extension in terms of the running example is given. Note that a matching extension M^* is always a valid matching: initially

$M^* = M$ holds and then $\{\Psi, \Psi^A\} \cap \{\Psi_1, \Psi_2\} = \emptyset$ is guaranteed in any subsequent iteration for each $\{\Psi_1, \Psi_2\} \in M^*$ before adding $\{\Psi, \Psi^A\}$ to M^*.

As each matching extension M^* is a perfect matching on the indistinguishability-graph for the set $psec^*$ of all potential secrets of M^*, each potential secret is in exactly one cluster of size 2. Moreover, in terms of availability, only a minimum number of additional potential secrets are created as a maximum matching M already covers as many potential secrets of the original policy $psec$ as possible.

3.3 Admissible Indistinguishabilities Based on Local Distortion

Until now, the clustering of policy elements is based on a purely abstract notion of admissible indistinguishabilities – which must be tailored to the needs of each specific application as argued in Sect. 3.1. An example for an easy to implement and moreover well-defined indistinguishability property, which locally restricts distortion within a disjunction, is the so-called *interchangeability*, which is applicable for *each* confidentiality policy consisting of ground atoms.

Definition 7 (Interchangeability). *The ground atoms $\Psi_1 = R(c_1, \ldots, c_n)$ and $\Psi_2 = R(d_1, \ldots, d_n)$ are* interchangeable*, if there is a* single differing position *$m \in \{1, \ldots, n\}$ with $c_m \neq d_m$ and $c_i = d_i$ for each $i \in \{1, \ldots, n\} \setminus \{m\}$. A set C of ground atoms over R is interchangeable, if all $\Psi_i, \Psi_j \in C$ with $\Psi_i \neq \Psi_j$ are pairwise interchangeable (and thus all differ at the same single position m).*

The indistinguishability-graph given in Fig. 3(a) is constructed based on the property of interchangeability and each of its edges is labeled with the single differing position of its incident potential secrets. Note that all indistinguishability-graphs resulting from this property have the structure known from a graph introduced by Knuth in [13] and further analyzed in [20], whose vertices are words of fixed length, which are neighbored if they differ in exactly one position.

A disjunction $\bigvee_{i \in \{1,\ldots,k\}} R(c_1, \ldots, c_{m-1}, \tilde{c}_m^{(i)}, c_{m+1}, \ldots, c_n)$ only consisting of pairwise interchangeable potential secrets has the advantage that the constant combinations of each of its disjuncts all only differ at the same single position m. Hence, it locally restricts distortion within this disjunction – and thus captures another aspect of maximizing availability – by providing definite information about all but the m-th columns in the sense that the original instance contains at least one tuple of the form $(c_1, \ldots, c_{m-1}, \square, c_{m+1}, \ldots, c_n)$ and by only hiding with which of the values $\tilde{c}_m^{(1)}, \ldots, \tilde{c}_m^{(k)}$ this tuple is combined.

If a total order with a successor function $succ(\cdot)$ is supposed to exist on the set Dom of constant symbols, the creation of an additional potential secret Ψ^A for an arbitrary potential secret $R(c_1, \ldots, c_n)$ is easy to define for the interchangeability property. Choose a differing position $m \in \{1, \ldots, n\}$ arbitrarily and initially set $\Psi^A := R(c_1, \ldots, \tilde{c}_m, \ldots, c_n)$ with $\tilde{c}_m := succ(c_m)$. As long as Ψ^A is in $psec$ or Ψ^A is equal to an already constructed additional potential secret, iteratively set $\tilde{c}_m := succ(\tilde{c}_m)$. Note that – demanding clusters of a minimum size of k^* – in a worst case scenario at most $(k^* - 1) \cdot |psec|$ additional constants are needed to create $k^* - 1$ additional potential secrets for each of the $|psec|$ many policy elements. Hence, this indistinguishability property is well-defined.

A disadvantage of this kind of indistinguishability clearly is that it only provides a suitable number of possible disjunctions if the majority of policy elements consist of constant combinations not differing much from each other. If this is not the case, a large number of additional potential secrets is needed and hence employing this kind of indistinguishability may result in a loss of availability. This is exemplified in Sect. 5 and demonstrates that the task of suitably defining admissible disjunctions crucially depends on the specific application considered.

4 Creation of Inference-Proof Weakenings

Before the overall algorithm creating an inference-proof weakening $weak(r, psec)$ of a complete database instance r and a confidentiality policy $psec$ can be developed, the construction of such a weakened instance must be defined. As motivated in Sect. 2, a weakened instance is a totally ordered sequence of sentences.

Definition 8 (Weakened Instance). *Suppose that r is a complete database instance over schema $\langle R|\mathcal{A}_R|\,\emptyset\,\rangle$ and $\mathcal{C}_r^*$ is an (extended) clustering of a confidentiality policy psec such that for each cluster $C \in \mathcal{C}_r^*$ there is a potential secret $\Psi \in C$ with $\mathcal{I}_r \models_M \Psi$. Then the incomplete* weakened instance *$weak(r, psec)$ is constructed as the following three totally ordered sequences of sentences of $\mathscr{L}$:*

- *(i)* Positive knowledge $weak(r, psec)^+$: *Each tuple $\boldsymbol{c} \in r$ with $R(\boldsymbol{c}) \not\models_{DB} \Psi$ for each $\Psi \in \bigcup_{C \in \mathcal{C}_r^*} C$ is modeled as a ground atom $R(\boldsymbol{c})$. All of these ground atoms are sorted lexicographically according to the order on Dom.*
- *(ii)* Disjunctive knowledge $weak(r, psec)^\vee$: *For each cluster $C \in \mathcal{C}_r^*$ the disjunction $\bigvee_{\Psi \in C} \Psi$ is constructed. First, for each of these disjunctions its disjuncts are sorted lexicographically according to the order on Dom and then all of these disjunctions are sorted in the same way.*
- *(iii)* Negative knowledge $weak(r, psec)^-$: *A completeness sentence (cf. Sect. 2) having a universally quantified variable X_j for each attribute $A_j \in \mathcal{A}_R$ is constructed. It has a disjunct $(\bigwedge_{i \in \{1,\ldots,n\} \text{ with } t_i \in Dom} X_i \equiv t_i)$ for each ground atom $R(t_1, \ldots, t_n)$ of $weak(r, psec)^+$ and for each (existentially quantified) atom*[1] *$(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ of a disjunction of $weak(r, psec)^\vee$. The above mentioned disjuncts are sorted in the same way as the disjunctions of $weak(r, psec)^\vee$. As a last disjunct $\neg R(X_1, \ldots, X_n)$ is added.*

An example of such a weakened instance is given in Fig. 4(c). Each weakened instance $weak(r, psec)$ contains only true information, i.e., $\mathcal{I}_r \models_M weak(r, psec)$, as for each ground atom $R(\boldsymbol{c})$ of $weak(r, psec)^+$ the tuple $\boldsymbol{c}$ is valid in r; each disjunction of $weak(r, psec)^\vee$ contains a disjunct Ψ_i with $\mathcal{I}_r \models_M \Psi_i$ by construction of $\mathcal{C}_r^*$; and for each constant combination $\boldsymbol{c} \in Dom^n$, for which $\neg R(\boldsymbol{c})$ holds by the completeness sentence of $weak(r, psec)^-$, the tuple $\boldsymbol{c}$ is invalid in r.

Now that all basic operations are known, the overall algorithm generating an inference-proof weakened instance is presented.

[1] This definition is generalized to be compatible to Sect. 6. If a potential secret is a ground atom, "$(\exists \boldsymbol{X})$" is dropped and each t_i is a constant symbol of Dom.

r	$+$	$-$
	(a,b,a)	(a,a,a)
	(a,b,b)	(a,a,b)
	(a,c,b)	$\vdots$
	(c,a,b)	

(a) Original instance r

$\{R(a,\underline{b},b),\ R(a,\underline{c},b)\ \}$,
$\{R(c,a,\underline{a}),\ R(c,a,\underline{b})^A\}$

(b) Clusters of a set $\mathcal{C}_r^*$ with a potential secret satisfied by $\mathcal{I}_r$

$R(a,b,a)$
$R(a,b,b) \vee R(a,c,b)$
$R(c,a,a) \vee R(c,a,b)$
$(\forall X)(\forall Y)(\forall Z)\ [$
$(X \equiv a\ \wedge\ Y \equiv b\ \wedge\ Z \equiv a)\ \vee$
$(X \equiv a\ \wedge\ Y \equiv b\ \wedge\ Z \equiv b)\ \vee$
$(X \equiv a\ \wedge\ Y \equiv c\ \wedge\ Z \equiv b)\ \vee$
$(X \equiv c\ \wedge\ Y \equiv a\ \wedge\ Z \equiv a)\ \vee$
$(X \equiv c\ \wedge\ Y \equiv a\ \wedge\ Z \equiv b)\ \vee$
$\neg R(X,Y,Z)\ ]$

(c) Weakening $\mathit{weak}(r, \mathit{psec})$ based on $\mathcal{C}_r^*$ obeying the policy of Fig. 3(a)

Fig. 4: Example of an inference-proof weakening obeying the policy of Fig. 3

Algorithm 1 (Inference-Proof Weakening). *Given a complete database instance r over $\langle R|\mathcal{A}_R|\ \emptyset\ \rangle$, a confidentiality policy psec of ground atoms of $\mathscr{L}$, a minimum size k^* of clusters and a well-defined notion of admissible indistinguishabilities, a weakened instance $\mathit{weak}(r, \mathit{psec})$ is created as follows:*

- **Stage 1** (independent of r): Disjoint clustering of potential secrets
  - *(i) Generate all admissible clusters with a minimum size of k^* (e.g., an indistinguishability-graph $G = (V, E)$ of psec (Def. 4))*
  - *(ii) Compute a disjoint clustering $\mathcal{C}$ based on the admissible clusters (e.g., a maximum matching $M \subseteq E$ on G (Def. 5))*
  - *(iii) Create $\mathcal{C}^*$ from $\mathcal{C}$ by extending each too small cluster of $\mathcal{C}$ to size k^* (e.g., by a matching extension M^* of M and psec (Def. 6))*
- **Stage 2** (dependent on r): Creation of weakened instance
  - *(iv) Create the subset $\mathcal{C}_r^* := \{\, C \in \mathcal{C}^* \mid \mathcal{I}_r \models_M \bigvee_{\Psi \in C} \Psi \,\}$ of (extended) clusters containing a potential secret not obeyed by $\mathcal{I}_r$*
  - *(v) Create the weakened instance $\mathit{weak}(r, \mathit{psec})$ based on r and $\mathcal{C}_r^*$ (Def. 8)*

An example of a weakened instance created by the *availability-maximizing* flavor of Algorithm 1 for the original instance of Fig. 4(a) is depicted in Fig. 4(c). The confidentiality policy, the corresponding indistinguishability-graph – constructed based on the *interchangeability* property – and the extended matching on which the set $\mathcal{C}_r^*$ of clusters given in Fig. 4(b) is based on is known from Fig. 3.

To understand the importance of *disjoint* clusters, consider the instance $r = \{\boldsymbol{c}_1\}$ and the *non-disjoint* clusters $C_1 = \{R(\boldsymbol{c}_1), R(\boldsymbol{c}_2)\}$ and $C_2 = \{R(\boldsymbol{c}_2), R(\boldsymbol{c}_3)\}$ with $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3 \in \mathit{Dom}^n$. Then, $\mathit{weak}(r, \mathit{psec})$ consists of $\mathit{weak}(r, \mathit{psec})^+ = \emptyset$ and $\mathit{weak}(r, \mathit{psec})^\vee = \{R(\boldsymbol{c}_1) \vee R(\boldsymbol{c}_2)\}$. Moreover, because of $R(\boldsymbol{c}_2) \vee R(\boldsymbol{c}_3) \notin \mathit{weak}(r, \mathit{psec})^\vee$ and by construction of the completeness sentence, an adversary knows $\mathcal{I}_r \not\models_M R(\boldsymbol{c}_2)$. Hence, he can infer that $\mathcal{I}_r \models_M \mathit{weak}(r, \mathit{psec})^\vee$ can only hold, if $\mathcal{I}_r \models_M R(\boldsymbol{c}_1)$, thereby violating the potential secret $R(\boldsymbol{c}_1)$ of C_1.

Theorem 1 (Inference-Proofness of Weakenings). *Given the inputs of Algorithm 1 (i.e., r over $\langle R|\mathcal{A}_R|\,\emptyset\,\rangle$, psec, k^*, and well defined indistinguishabilities), this algorithm generates an inference-proof weakened instance $weak(r, psec)$ such that for each potential secret $\Psi \in psec$ the existence of a complete alternative instance r^{Ψ} over $\langle R|\mathcal{A}_R|\emptyset\rangle$ is guaranteed. This alternative instance r^{Ψ} obeys Ψ, i.e., $\mathcal{I}_{r^{\Psi}} \not\models_M \Psi$, and the weakening $weak(r^{\Psi}, psec)$ generated by Algorithm 1 is indistinguishable from $weak(r, psec)$, i.e., $weak(r^{\Psi}, psec) = weak(r, psec)$.*

Proof. Consider an arbitrary potential secret $\tilde{\Psi} \in psec$ and suppose that Stage 1 generated a (possibly extended) disjoint clustering $\mathcal{C}^*$ with clusters of a minimum size of $k^* \geq 2$. Assume that $\tilde{\Psi}$ is in the cluster $\tilde{C} = \{\tilde{\Psi}, \tilde{\Psi}_{I_1}, \ldots, \tilde{\Psi}_{I_{k-1}}\} \in \mathcal{C}^*$.

If $\mathcal{I}_r \not\models_M \bigvee_{\Psi\in\tilde{C}} \Psi$, the complete alternative instance $r^{\tilde{\Psi}}$ is r itself, i.e., $r^{\tilde{\Psi}} := r$. This implies $\mathcal{I}_{r^{\tilde{\Psi}}} \not\models_M \bigvee_{\Psi\in\tilde{C}} \Psi$ and consequently $r^{\tilde{\Psi}}$ obeys $\tilde{\Psi}$, i.e., $\mathcal{I}_{r^{\tilde{\Psi}}} \not\models_M \tilde{\Psi}$, because of $\mathcal{I}_{r^{\tilde{\Psi}}} \models_M \neg(\bigvee_{\Psi\in\tilde{C}} \Psi) = \bigwedge_{\Psi\in\tilde{C}} (\neg\Psi)$. As a direct consequence of $r^{\tilde{\Psi}} = r$ the property of indistinguishability holds, i.e., $weak(r^{\tilde{\Psi}}, psec) = weak(r, psec)$.

If $\mathcal{I}_r \models_M \bigvee_{\Psi\in\tilde{C}} \Psi$ with $\tilde{\Psi} = R(\boldsymbol{c}_{\tilde{\Psi}}) \in \tilde{C}$ and a $\tilde{\Psi}_{I_m} = R(\boldsymbol{c}_{\tilde{\Psi}_{I_m}}) \in \tilde{C}$, the complete alternative instance is $r^{\tilde{\Psi}} := (r \setminus \{\boldsymbol{c}_{\tilde{\Psi}}\}) \cup \{\boldsymbol{c}_{\tilde{\Psi}_{I_m}}\}$. Hence, $r^{\tilde{\Psi}}$ obeys $\tilde{\Psi}$, i.e., $\mathcal{I}_{r^{\tilde{\Psi}}} \not\models_M \tilde{\Psi}$, and $\mathcal{I}_{r^{\tilde{\Psi}}} \models_M \bigvee_{\Psi\in\tilde{C}} \Psi$ because of $\mathcal{I}_{r^{\tilde{\Psi}}} \models_M \tilde{\Psi}_{I_m}$. For each other cluster $C \in M^*$ with $C \neq \tilde{C}$ the corresponding disjunction $\bigvee_{\Psi\in C} \Psi$ is satisfied by $\mathcal{I}_{r^{\tilde{\Psi}}}$ if and only if it is satisfied by $\mathcal{I}_r$ because of $r^{\tilde{\Psi}} \setminus \{\boldsymbol{c}_{\tilde{\Psi}}, \boldsymbol{c}_{\tilde{\Psi}_{I_m}}\} = r \setminus \{\boldsymbol{c}_{\tilde{\Psi}}, \boldsymbol{c}_{\tilde{\Psi}_{I_m}}\}$ and because of $\tilde{\Psi} \notin C$ and $\tilde{\Psi}_{I_m} \notin C$ by the disjoint clustering.

This implies $\mathcal{C}^*_{r^{\tilde{\Psi}}} = \mathcal{C}^*_r$ and hence also $weak(r^{\tilde{\Psi}}, psec)^{\vee} = weak(r, psec)^{\vee}$. As $r^{\tilde{\Psi}}$ and r only differ in $\boldsymbol{c}_{\tilde{\Psi}}$ and $\boldsymbol{c}_{\tilde{\Psi}_{I_m}}$ and as $\tilde{C}$ with $R(\boldsymbol{c}_{\tilde{\Psi}}), R(\boldsymbol{c}_{\tilde{\Psi}_{I_m}}) \in \tilde{C}$ is a cluster of both $\mathcal{C}^*_{r^{\tilde{\Psi}}}$ and $\mathcal{C}^*_r$, also $weak(r^{\tilde{\Psi}}, psec)^{+} = weak(r, psec)^{+}$ holds. By construction of the completeness sentence, $weak(r^{\tilde{\Psi}}, psec)^{-} = weak(r, psec)^{-}$ directly follows and so the property of indistinguishability, i.e., $weak(r^{\tilde{\Psi}}, psec) = weak(r, psec)$, holds, provided that the sentences of both of these sequences are arranged in the same order. □

5 Efficiency of the Approach

After developing Algorithm 1, a prototype implementation of the *availability-maximizing* instantiation of this algorithm (cf. Sect. 3.2) is now sketched and evaluated theoretically as well as experimentally. Thereby *interchangeability* (cf. Def. 7) is employed as a well-defined indistinguishability property.

Within Stage 1 of Algorithm 1 the indistinguishability-graph is constructed efficiently with a flavor of the merge-join algorithm (cf. Sect. 3.1), which is well-known from relational databases [19]. In typical scenarios the runtime of this algorithm is significantly better than its worst-case complexity $O(|psec|^2)$ [19].

To next compute a maximum matching (cf. [14,17]), the prototype benefits from the "Boost"-library [8]. Although a maximum matching on a general graph $G = (V, E)$ can be computed in $O(\sqrt{|V|}\cdot|E|)$ (cf. [22]), common implementations

as provided by "LEDA" [18] or "Boost" [8] prefer an algorithm performing in $O(|V| \cdot |E| \cdot \alpha(|E|, |V|))$ with $\alpha(|E|, |V|) \leq 4$ for any feasible input.

Stage 1 finally computes a matching extension M^* and in a worst-case scenario $|psec|$ different additional potential secrets – whose creation in the case of interchangeability is sketched in Sect. 3.3 – are needed. Provided that binary search is employed to check collisions of tentatively constructed additional potential secrets, M^* is constructed in $O(|psec|^2 \cdot \log(|psec|))$. But note that this upper bound is purely theoretic and usually not even approached.

Stage 2 of Algorithm 1 first creates the subset $\mathcal{C}_r^*$ of clusters based on M^* in $O(|psec| \cdot \log(|r|))$ by employing binary search to check which potential secrets in the form of ground atoms are satisfied by the original instance. Finally, the weakened instance is constructed. Again using binary search, $weak(r, psec)^+$ is constructed in $O(|r| \cdot \log(|psec|))$ and sorted in $O(|r| \cdot \log(|r|))$; $weak(r, psec)^\vee$ is constructed in $O(|psec|)$ and sorted in $O(|psec| \cdot \log(|psec|))$; and $weak(r, psec)^-$ is constructed in $O(|r| + |psec|)$ and sorted in $O((|r| + |psec|) \cdot \log(|r| + |psec|))$.

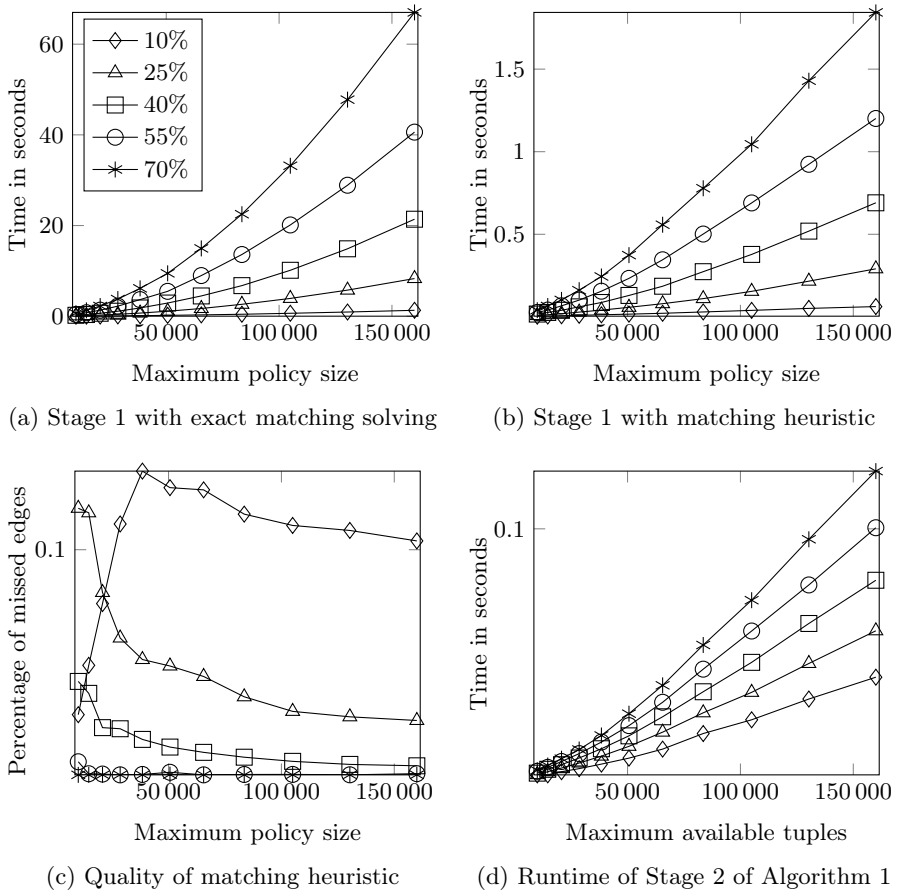
The prototype is implemented in Java 7, except for the C++ implementation of the matching algorithm (see above). All experiments were run under Ubuntu 14.04 on an "Intel Core i7-4770" machine with 32 GB of main memory and each published result is based on the average results of 100 experiments.

To generate the input data for a first test setup, for each experiment a particular finite set $\mathcal{D} \subseteq Dom$ of constant symbols is available for the construction of the constant combinations of all database tuples and potential secrets, which are all supposed to be of arity 4. As the cardinality of $\mathcal{D}$ varies over the experiments from $|\mathcal{D}| = 10$ to $|\mathcal{D}| = 20$, the cardinality of the set $constComb(\mathcal{D}) := \mathcal{D}^n$ of all possible constant combinations varies from $10^4 = 10\,000$ to $20^4 = 160\,000$.

To evaluate Stage 1 of Algorithm 1, for each of the possible cardinalities of $\mathcal{D}$ a randomly chosen subset of $constComb(\mathcal{D})$ is selected to construct a random confidentiality policy $psec$ as input data for an experiment. Thereby, the fraction of tuples of $constComb(\mathcal{D})$ contained in the policy is stepwise increased from 10% to 70% of all tuples of $constComb(\mathcal{D})$. Hence, the average vertex degree of the corresponding indistinguishability-graphs is also stepwise increased.

As depicted in Fig. 5(a), even for large policies Stage 1 of Algorithm 1 performs very well in constructing clusterings of the policies. If an even faster computation is needed, the matching heuristic presented in [17] – which performs in time linear to the size of the graph – can be employed. As depicted in Fig. 5(b), the usage of this heuristic significantly improves the runtime of Stage 1 and usually looses only a negligible fraction of matching edges in relation to a optimum solution, as demonstrated in Fig. 5(c). Hence, using this heuristic results only in a slight loss of availability, as an additional potential secret is needed for each vertex uncovered by the matching.

To evaluate Stage 2 of Algorithm 1, for each of the possible cardinalities of $\mathcal{D}$ two randomly chosen subsets of $constComb(\mathcal{D})$ are selected to construct a random database instance r as well as a random confidentiality policy $psec$. The fraction of tuples of $constComb(\mathcal{D})$ contained in r is stepwise increased from 10% to 70% of all tuples of $constComb(\mathcal{D})$ while the fraction of tuples contained in



(a) Stage 1 with exact matching solving

(b) Stage 1 with matching heuristic

(c) Quality of matching heuristic

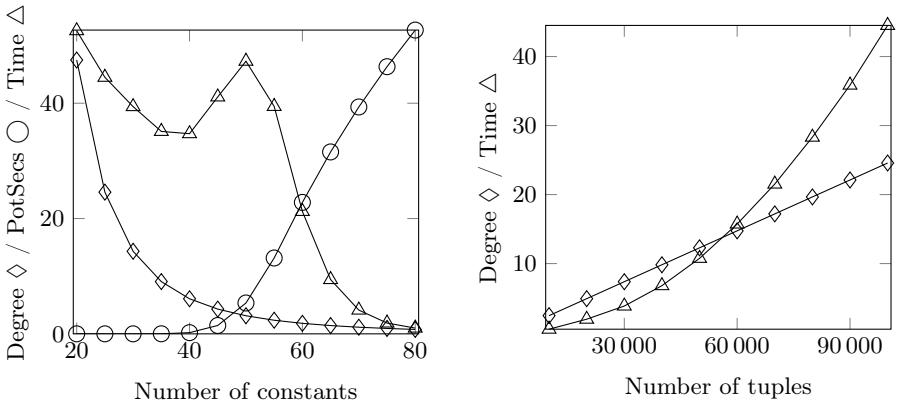
(d) Runtime of Stage 2 of Algorithm 1

Fig. 5: Experimental evaluation of Algorithm 1 for the first test setup

psec is fixed to 40%. According to Fig. 5(d), the runtime of Stage 2 needed to construct a weakening based on a given clustering is negligible.

At first glance, input instances constructed based on 20 or even just 10 available constants might look like "toy examples", but note that for a clustering of fully random potential secrets based on the interchangeability property these instances are the expensive inputs: the relatively small number of available constants leads to constant combinations which are likely not to differ much from each other and hence the corresponding indistinguishability-graphs have a large number of edges making the computation of a maximum matching expensive.

This is demonstrated by experiments always constructing 100 000 fully random tuples for a number of available constants varying from 20 to 80. As shown in Fig. 6(a), increasing the number of available constants leads to a decreasing of the average vertex degree of the indistinguishability-graphs. In the end the



(a) Average vertex *degree* of graph, number of additional *potential secrets* in thousands, and *runtime* of Stage 1 in seconds for 100 000 random tuples of arity 4 constructed for a varying number of constants

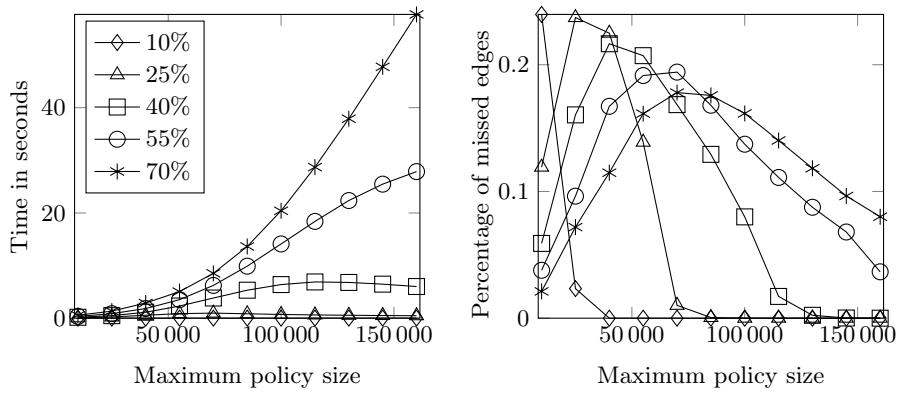
(b) Average vertex *degree* of graph, and *runtime* of Stage 1 in seconds for a fixed number of 25 constants and for a number of tuples of arity 4 varying from 10 000 to 100 000

Fig. 6: Evaluation of the interchangeability property within the first test setup

graphs decompose into a large number of small connected components and as hence the clustering becomes trivial the runtime of Stage 1 also declines. These results are also verified by a second experiment fixing the number of constants to 25 and linearly increasing the number of constructed potential secrets from 10 000 to 100 000. As shown in Fig. 6(b), this leads to an also linearly increasing average vertex degree while the runtime of Stage 1 increases much stronger.

As very low average vertex degrees moreover lead to a large number of additional potential secrets (plotted in thousands in Fig. 6(a)), the interchangeability property only provides suitably high availability, if the majority of policy elements consist of constant combinations not differing much from each other. This demonstrates that the task of finding a suitable notion of admissible indistinguishabilities crucially depends on the specific application considered.

Next, a second test setup is initiated, which is supposed to be more practical than the fully random setup. This second setup – only considering Stage 1 as the runtime of Stage 2 is now known to be negligible – is based on a set of objects, each of which has two attributes: the first attribute has a domain, whose cardinality k is stepwise increased from 2 to 32, and the second attribute has a domain of cardinality 100. Considering binary relations between some of these objects, each constructible object is paired with 50 randomly chosen other constructible objects – resulting in $k \cdot 100 \cdot 50$ tuples of arity 4, i.e., the number of available constant combinations again varies from 10 000 (for $k = 2$) to 160 000 (for $k = 32$). Similarly to the first test setup, for each value of k the confidentiality policy is created as a randomly chosen subset of all available constant combinations, whose cardinality is stepwise increased from 10% to 70%.



(a) Stage 1 with exact matching solving (b) Quality of matching heuristic

Fig. 7: Experimental evaluation of Algorithm 1 for the second test setup

For this second test setup the exact computation of Stage 1 performs better than using the first test setup (cf. Fig. 7(a)) as the resulting graphs have a lower but non-trivial average vertex degree. The runtime of the heuristic computation is as good as known from the fully random setup, but the number of lost matching edges is slightly higher compared to the first test setup (cf. Fig. 7(b)) as the graphs resulting from the second test setup often have a lower but non-trivial average vertex degree leading to slightly weaker but still very decent results.

6 Extending the Approach

So far, only potential secrets in the form of ground atoms have been considered. To improve the expressiveness of confidentiality policies, potential secrets are from now on so-called *existentially quantified atoms* known from [5]. Intuitively, an existentially quantified potential secret $\Psi = (\exists Z)\, R(a, b, Z)$ states that an adversary *must not* get to know that a tuple $(a, b, \tilde{c})$ with an arbitrary constant symbol $\tilde{c} \in \mathit{Dom}$ is valid in the original instance r considered.

Definition 9 (Existentially Quantified Atom). *A sentence of $\mathscr{L}$ is an* existentially quantified atom *if it is of the form $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and*

(i) each term t_i is either a constant symbol of Dom or a variable of $\boldsymbol{X}$,
(ii) the set $\boldsymbol{X}$ of existentially quantified variables is $\boldsymbol{X} = \{t_1, \ldots, t_n\} \setminus \mathit{Dom}$,
(iii) each variable can only occur once, i.e., $t_i \neq t_j$ for all $t_i, t_j \in \boldsymbol{X}$ with $i \neq j$.

Though implication is generally hard (if not even impossible) to decide within first-order logic [4], under DB-Semantics (cf. Def. 1) it is easy to decide for existentially quantified atoms [5]: $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n) \models_{DB} (\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ iff for each term $\bar{t}_i$, which is a constant symbol of Dom, the term t_i is also a constant symbol of Dom such that $t_i = \bar{t}_i$.

Now, suppose that an instance $r = \{R(a,b,c)\}$ and a confidentiality policy $psec = \{(\exists Z)\, R(a,b,Z),\ (\exists Z)\, R(b,b,Z),\ R(a,b,c),\ R(a,b,d)\}$ are inputs for a flavor of Algorithm 1 creating the clusters $C_1 = \{(\exists Z)\, R(a,b,Z),\ (\exists Z)\, R(b,b,Z)\}$ and $C_2 = \{R(a,b,c),\ R(a,b,d)\}$. The weakened instance $weak(r, psec)$ then contains the disjunction $R(a,b,\underline{c}) \vee R(a,b,\underline{d})$ and hence directly implies the knowledge $(\exists Z)\, R(a,b,Z)$ which itself is protected by a potential secret of $psec$.

The preceding example indicates that this flavor of Algorithm 1 could create a weakened instance which contains disjunctions implying knowledge protected by potential secrets. So, this implied (and hence weaker) knowledge is still too strong. To avoid the construction of too strong disjunctions, the algorithm must clean the given confidentiality policy in a preprocessing step, i.e., the policy is reduced to its weakest sentences. Moreover, adding the constructed additional potential secrets to this set must not violate the properties of a cleaned set.

Definition 10 (Cleaned Set). *Let $\mathcal{S}$ be a set of sentences of $\mathscr{L}$. Its* cleaned set *$\hat{\mathcal{S}}$ is a maximum subset of weakest sentences of $\mathcal{S}$ such that no pair of different sentences of $\hat{\mathcal{S}}$ is semantically equivalent. $\Psi \in \mathcal{S}$ is a* weakest sentence *of $\mathcal{S}$, if for each sentence $\Psi' \in \mathcal{S}$ either $\Psi' \models_{DB} \Psi$ or both $\Psi' \not\models_{DB} \Psi$ and $\Psi \not\models_{DB} \Psi'$.*

Reconsidering the example, $\widehat{psec} = \{(\exists Z)\, R(a,b,Z),\ (\exists Z)\, R(b,b,Z)\}$ is the cleaned policy. Assuming that $\{\widehat{psec}\}$ is the created clustering, the weakening $weak(r, \widehat{psec})$ only contains the disjunction $(\exists Z)\, R(a,b,Z) \vee (\exists Z)\, R(b,b,Z)$ not implying any (weaker) knowledge which itself is protected.

In particular, even the potential secrets $R(a,b,c)$ and $R(a,b,d)$ only contained in the original policy $psec$ are protected by $weak(r, \widehat{psec})$: from an adversary's point of view an alternative instance r' with $\mathcal{I}_{r'} \models_M weak(r, \widehat{psec})$ and $\mathcal{I}_{r'} \not\models_M (\exists Z)\, R(a,b,Z)$ is possible and for this instance also $\mathcal{I}_{r'} \not\models_M R(a,b,c)$ and $\mathcal{I}_{r'} \not\models_M R(a,b,d)$ holds. This implicit protection of all removed policy elements $psec \setminus \widehat{psec}$ by the cleaned policy $\widehat{psec}$ can be generalized as follows.

Lemma 1 (Implicit Protection). *Let Ψ_S and Ψ_W be sentences of $\mathscr{L}$ such that Ψ_W is weaker than Ψ_S, i.e., $\Psi_S \models_{DB} \Psi_W$, and let $\mathcal{I}_r$ be a DB-Interpretation with $\mathcal{I}_r \not\models_M \Psi_W$. Then Ψ_S is not satisfied by $\mathcal{I}_r$ either, i.e., $\mathcal{I}_r \not\models_M \Psi_S$.*

In many real-world scenarios an adversary is supposed to also have some *a priori knowledge* in addition to the knowledge provided by the database (cf. [3]). A priori knowledge is then modeled as a finite set *prior* of sentences of $\mathscr{L}$ and usually includes the set SC_R of semantic constraints (cf. Sect. 2), i.e., $SC_R \subseteq prior$. All sentences of *prior* are supposed to be satisfied by the original instance r, i.e., $\mathcal{I}_r \models_M prior$, and furthermore do not directly compromise the confidentiality policy *psec*, i.e., $prior \not\models_{DB} \Psi$ for each potential secret $\Psi \in psec$. To make a first step towards the handling of a priori knowledge, an adversary is now supposed to be also aware of such a set *prior* of ground atoms of $\mathscr{L}$.

Similar to Def. 3, a notion of admissible indistinguishabilities might require that for some potential secrets of the cleaned policy $\widehat{psec}$ up to $k^* - 1$ additional potential secrets can be constructed. To moreover ensure that all non-implications provided by cleaning the policy are not affected by combinatorial

effects, the domain *Dom* must contain at least one "fresh" constant symbol not occurring in a potential secret of *psec* or a constructed additional potential secret. In terms of the credibility of these non-implications from an adversary's point of view, a much larger supply of these "fresh" constant symbols is of course highly desirable.

Definition 11 (Well-Defined Indistinguishability Ext.). *Given a cleaned confidentiality policy* $\widehat{psec}$*, an adversary's a priori knowledge prior, the domain Dom of* $\mathscr{L}$ *and a minimum size* k^* *of clusters, a notion of admissible indistinguishabilities is* well-defined*, if there is a set* $\mathcal{C}^*$ *such that*

(i) for each $\Psi \in \widehat{psec}$ *the set* $\mathcal{C}^*$ *contains a cluster* $C_\Psi = \{\Psi, \Psi_{I_1}, \ldots, \Psi_{I_{k^*-1}}\}$ *(possibly extended) such that* $\Psi \neq \Psi_{I_i}$ *for* $1 \leq i \leq k^* - 1$ *and* $\Psi_{I_i} \neq \Psi_{I_j}$ *for* $1 \leq i < j \leq k^* - 1$ *and* $\bigvee_{\bar{\Psi} \in C_\Psi} \bar{\Psi}$ *is an admissible indistinguishability,*
(ii) $C_\Psi \cap C_{\Psi'} = \emptyset$ *holds for all clusters* $C_\Psi, C_{\Psi'} \in \mathcal{C}^*$ *with* $C_\Psi \neq C_{\Psi'}$*,*
(iii) $\bigcup_{C \in \mathcal{C}^*} C$ *is a cleaned set,*
(iv) $prior \not\models_{DB} \Psi^A$ *for each (additional)* Ψ^A *of* $\mathcal{C}^*$ *with* $\Psi^A \notin \widehat{psec}$*,*
(v) there is a deterministic algorithm creating each (additional) Ψ^A *of* $\mathcal{C}^*$ *with* $\Psi^A \notin \widehat{psec}$*, thereby (finitely) augmenting the active domain of psec,*
(vi) the active domain of $\mathcal{C}^*$ *is contained in Dom and*
(vii) Dom contains at least one constant not in the active domain of $\mathcal{C}^*$*.*

As a direct consequence of this extension of Def. 3, no well-defined notion of indistinguishability can be found, if the policy *psec* contains a potential secret Ψ_W which is semantically equivalent to the weakest possible potential secret $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$ without any constant symbols. In this case the cleaned policy $\widehat{psec}$ only contains Ψ_W and no additional potential secret Ψ_W^A can be found for Ψ_W as $\{\Psi_W, \Psi_W^A\}$ cannot be a cleaned set because of $\Psi_W^A \models_{DB} \Psi_W$.

Based on the thoughts presented so far, Algorithm 1 can be extended. Its inference-proofness can be basically proved as known from Theorem 1, but each "secure" alternative instance must furthermore satisfy an adversary's a priori knowledge to be credible from this adversary's point of view [3].

Theorem 2 (Inference-Proofness of Weakenings). *Let* r *be a complete instance over* $\langle R|\mathcal{A}_R|SC_R\rangle$*; psec be a policy of existentially quantified atoms;* k^* *be the minimum size of clusters; and assume that a well-defined notion of indistinguishabilities is given. Moreover, prior (with* $SC_R \subseteq prior$*) is a priori knowledge of ground atoms such that* $\mathcal{I}_r \models_M prior$ *and* $prior \not\models_{DB} \Psi$ *for each* $\Psi \in psec$*.*

The extended algorithm then creates an inference-proof weakened instance[2] $weak(r, psec)$ *such that for each potential secret* $\Psi \in psec$ *the existence of a complete alternative instance* r^Ψ *over* $\langle R|\mathcal{A}_R|SC_R\rangle$ *is guaranteed. This alternative instance* r^Ψ *obeys* Ψ*, i.e.,* $\mathcal{I}_{r^\Psi} \not\models_M \Psi$*, satisfies the a priori knowledge prior, i.e.,* $\mathcal{I}_{r^\Psi} \models_M prior$*, and the weakening* $weak(r^\Psi, psec)$ *is indistinguishable from* $weak(r, psec)$*, i.e.,* $weak(r^\Psi, psec) = weak(r, psec)$*.*

The detailed proof of Theorem 2 is omitted for lack of space.

[2] Though the weakening of an instance now also depends on *prior*, for convenience the weakening-operator $weak(\cdot, \cdot)$ is not extended to explicitly reflect this third input.

7 Conclusion and Future Work

We developed a generic approach provably protecting sensitive information specified by a confidentiality policy consisting of ground atoms – even if an adversary employs inferences. This is achieved by weakening a database instance by means of disjunctions. Furthermore, an algorithm for an availability-maximizing flavor of this approach has been proposed and an implementation of this algorithm based on interchangeability has been shown to be highly efficient. Moreover, the generic approach has also been extended to protect more expressive confidentiality policies while also considering an adversary's a priori knowledge.

But a priori knowledge restricted to ground atoms does not allow for modeling commonly used semantic database constraints such as the well-known classes of Equality Generating and Tuple Generating Dependencies (cf. [1]). Examples for achieving inference-proofness under versatile subclasses of these semantic constraints are given in [6,7] and should be transferred to the current approach.

Moreover, the definition of inference-proofness underlying this work only guarantees the existence of at least one "secure" alternative instance from an adversary's point of view (cf. Theorem 1 and Theorem 2). But in terms of enhancing confidentiality it might be desirable to strengthen this definition to always guarantee a certain number k of different "secure" alternative instances. As discussed for the generic approach, this can be achieved by increasing the length of disjunctions (cf. Sect. 3.1). Hence, algorithms constructing availability-maximizing clusters of size ≥ 3 should be developed on the operational level.

As known from Sect. 3.3, each disjunction of pairwise interchangeable disjuncts preserves definite information about all but one position of each ground atom and *generalizes* each distorted value to a wider set of possible values. This idea of generalizing values is similarly used for k-anonymization and ℓ-diversification [10,16,21]. So, it might be worthwhile to extend our approach to deal with confidentiality policies already containing disjunctions and to then model k-anonymization and ℓ-diversification within such an extension.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading (1995)
2. Bertossi, L.E., Li, L.: Achieving data privacy through secrecy views and null-based virtual updates. IEEE Transactions on Knowledge and Data Engineering 25(5), 987–1000 (2013)
3. Biskup, J.: Inference-usability confinement by maintaining inference-proof views of an information system. International Journal of Computational Science and Engineering 7(1), 17–37 (2012)
4. Biskup, J., Bonatti, P.A.: Controlled query evaluation with open queries for a decidable relational submodel. Annals of Mathematics and Artificial Intelligence 50(1–2), 39–77 (2007)
5. Biskup, J., Hartmann, S., Link, S., Lochner, J.H., Schlotmann, T.: Signature-based inference-usability confinement for relational databases under functional and join

dependencies. In: Cuppens-Boulahia, N., Cuppens, F., García-Alfaro, J. (eds.) Data and Applications Security and Privacy XXVI, DBSec 2012. LNCS, vol. 7371, pp. 56–73. Springer, Heidelberg (2012)
6. 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 – 27th Annual IFIP WG 11.3 Conference, DBSec 2013. LNCS, vol. 7964, pp. 17–32. Springer, Heidelberg (2013)
7. Biskup, J., Wiese, L.: A sound and complete model-generation procedure for consistent and confidentiality-preserving databases. Theoretical Computer Science 412(31), 4044–4072 (2011)
8. Boost Graph Library: Maximum cardinality matching (2014), `http://www.boost.org/doc/libs/1_55_0/libs/graph/doc/maximum_matching.html`
9. Campan, A., Truta, T.M.: Data and structural k-anonymity in social networks. In: Bonchi, F., Ferrari, E., Jiang, W., Malin, B. (eds.) Privacy, Security, and Trust in KDD, PinKDD 2008. LNCS, vol. 5456, pp. 33–54. Springer, Heidelberg (2008)
10. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Samarati, P.: k-Anonymity. In: Yu, T., Jajodia, S. (eds.) Secure Data Management in Decentralized Systems, Advances in Information Security, vol. 33, pp. 323–353. Springer, New York, NY (2007)
11. Fung, B.C., Wang, K., Fu, A.W.C., Yu, P.S.: Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques. Data Mining and Knowledge Discovery, CRC Press, Boca Raton, FL (2011)
12. Hay, M., Miklau, G., Jensen, D., Towsley, D.F., Li, C.: Resisting structural re-identification in anonymized social networks. VLDB Journal 19(6), 797–823 (2010)
13. Knuth, D.E.: The Stanford GraphBase: A Platform for Combinatorial Computing. ACM Press, New York, NY (1993)
14. Korte, B., Vygen, J.: Combinatorial Optimization: Theory and Algorithms. Algorithms and Combinatorics, Springer, Heidelberg, 5th edn. (2012)
15. Levesque, H.J., Lakemeyer, G.: The Logic of Knowledge Bases. The MIT Press, Cambridge, MA (2000)
16. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: ℓ-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data 1(1) (2007)
17. Magun, J.: Greedy matching algorithms: An experimental study. ACM Journal of Experimental Algorithmics 3(6) (1998)
18. Mehlhorn, K., Näher, S.: LEDA: A platform for combinatorial and geometric computing. Cambridge University Press, Cambridge (1999)
19. Ramakrishnan, R., Gehrke, J.: Database Management Systems. McGraw-Hill, Boston, MA, 3rd edn. (2003)
20. Stiege, G.: Playing with Knuth's words.dat. Tech. Rep. 1/12, Department of Computer Science, University of Oldenburg, Germany (May 2012)
21. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5), 557–570 (2002)
22. Vazirani, V.V.: A theory of alternating paths and blossoms for proving correctness of the $O(\sqrt{|V|} \cdot |E|)$ general graph maximum matching algorithm. Combinatorica 14(1), 71–109 (1994)
23. Wong, R.C.W., Fu, A.W.C.: Privacy-Preserving Data Publishing – An Overview. Synthesis Lectures on Data Management, Morgan & Claypool Publishers, San Rafael, CA (2010)