technische universität
dortmund

Inference-Proof Materialized Views

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Marcel Preuß

Dortmund

2016

Tag der mündlichen Prüfung: 24.08.2016

Dekan: Prof. Dr.-Ing. Gernot Fink

1. Gutachter: Prof. Dr. Joachim Biskup
2. Gutachterin: Prof. Dr. Gabriele Kern-Isberner

Science never solves a problem
without creating ten more.

George Bernard Shaw

This thesis is typeset using LaTeX.

Foreword

At this point I would like to take the opportunity to express my thanks to all those people, who accompanied and supported me during my academic career and, in particular, during the writing of this thesis.

First of all, I would like to gratefully thank Joachim Biskup, who actively supervised my research, on which this thesis is based. We started to work together in the year 2006, when Joachim offered me a job as a student assistant. After finishing my diploma thesis under his supervision in the year 2010, I then started my professional career as a research assistant at his chair. During my joint work with Joachim, I always appreciated our fruitful discussions, which greatly benefited from Joachim's scientific expertise and his far-sighted view on approaches to solve scientific research problems.

My thanks also go to Gabriele Kern-Isberner for reviewing this thesis and for giving me helpful hints how to improve the manuscript of this thesis as well as to Jens Teubner and Lars Hildebrand for their willingness to complement the committee for my doctoral examination.

Furthermore, I would like to thank all my current and former colleagues, who contributed to the feeling of well-being at our chair. In particular, I would like to mention Ralf Menzel, who often turned out to be a valuable discussion partner to sort my thoughts when solving open research problems or struggling with LaTeX, and Jens Teubner, who welcomed our research group as the new chair holder after Joachim's formal retirement.

Last but not least, I would like to thank my family, and above all my parents Petra and Manfred and my girlfriend Patricia, who have supported me with their understanding and their active help.

Abstract

Nowadays, publishing of data is ubiquitous, but usually only permitted when complying with a confidentiality policy to respect privacy or other secrecy concerns. To this end, this thesis proposes an approach to weaken an original database instance to a weakened view on this instance. This view is inference-proof in the sense of "Controlled Interaction Execution" and does hence provably *not* enable an adversary to infer confidential knowledge – even if this adversary tries to deduce confidential knowledge on the basis of a released weakened view, his general awareness of the protection mechanism and some a priori knowledge he might possibly have about the original database instance or the world in general.

To achieve this goal within a logic-oriented modeling, all pieces of definite knowledge that compromise an element of a confidentiality policy are (whenever possible) replaced by weaker but true disjunctions of policy elements. Although this disjunctive knowledge deliberately introduces uncertainty about confidential knowledge, it still provides more information about the original database instance than complete refusals of confidential knowledge. To further guarantee that all of these weakening disjunctions are – with respect to a considered application scenario – both credible in terms of confidentiality and meaningful in terms of availability, a criterion specifying which policy elements might possibly be grouped together to an admissible weakening disjunction can be defined.

This approach is first developed in a *generic* way in the sense that non-trivial disjunctions of any length ≥ 2 might be employed and the achieved level of confidentiality varies with the length of disjunctions. Thereby, all knowledge is modeled within a restricted but expressive subclass of first-order logic, which allows for efficient decisions on the validity of implication relationships without general theorem proving. Afterwards, an *availability-maximizing* instantiation of this generic approach is presented, which aims at constructing disjunctions of length 2 efficiently on the basis of graph clustering, and is then also extended to handle an adversary's a priori knowledge in the form of a restricted subclass of well-known "Tuple Generating Dependencies" without losing its inference-proofness or efficiency.

To demonstrate the practical efficiency of this (extended) availability-maximizing approach, a prototype implementation is developed and evaluated under different experiment setups. Thereby, disjunctions are constructed on the basis of an admissibility criterion, which (locally) maximizes availability within a disjunction in the sense that both of its disjuncts differ in only one constant parameter and thereby generalizes this constant parameter to a wider set of possible values.

Zusammenfassung

Obwohl die Veröffentlichung von Daten heutzutage allgegenwärtig ist, ist diese häufig nur dann gestattet, wenn dabei Vertraulichkeitsanforderungen beachtet werden. Vor diesem Hintergrund wird in dieser Arbeit ein Ansatz entwickelt, um abgeschwächte Sichten auf gegebene Datenbankinstanzen zu erzeugen. Eine solche abgeschwächte Sicht ist dabei inferenzsicher im Sinne der sogenannten „Kontrollierten Interaktionsauswertung“ und verhindert damit beweisbar, dass ein Angreifer vertrauliche Information erlangen kann – selbst dann, wenn dieser Angreifer versucht, diese Information unter Zuhilfenahme seiner Kenntnis über den Sicherheitsmechanismus und etwaigem Vorwissen über die Datenbankinstanz oder allgemeine Sachverhalte logisch zu erschließen.

Dieses Ziel wird innerhalb einer logik-orientierten Modellierung verwirklicht, in der alles sichere Wissen, das die Vertraulichkeitspolitik verletzt, (soweit möglich) durch schwächere, aber dennoch wahre Disjunktionen bestehend aus Elementen der Vertraulichkeitspolitik ersetzt wird. Auch wenn dieses disjunktive Wissen bewusst Unsicherheit über vertrauliche Information erzeugt, stellt es dennoch mehr Information als eine vollständige Geheimhaltung von vertraulicher Information bereit. Um dabei sicherzustellen, dass Disjunktionen im Hinblick auf ein betrachtetes Einsatzszenario sowohl glaubwürdig als auch aussagekräftig sind, kann ein Kriterium definiert werden, aus welchen Kombinationen von Elementen der Vertraulichkeitspolitik eine mögliche Disjunktion bestehen kann.

Dieser Ansatz wird erst in einer generischen Variante entwickelt, in der nicht-triviale Disjunktionen jeder Länge ≥ 2 zum Einsatz kommen können und das erreichte Maß an Vertraulichkeit mit der Länge der Disjunktionen variiert. Dabei wird jegliches Wissen in einem eingeschränkten, aber dennoch vielfältig einsetzbaren Fragment der Prädikatenlogik modelliert, in dem die Gültigkeit von Implikationsbeziehungen effizient ohne den Einsatz von Theorembeweisern entschieden werden kann. Anschließend wird eine Variante dieses generischen Ansatzes vorgestellt, die die Verfügbarkeit maximiert, indem Disjunktionen der Länge 2 effizient mit Hilfe von Clustering auf Graphen konstruiert werden. Diese Variante wird daraufhin derart erweitert, dass sie auch dann effizient inferenzsichere Sichten erzeugen kann, wenn ein Angreifer Vorwissen in Form einer eingeschränkten Unterklasse von sogenannten „Tuple Generating Dependencies“ hat.

Um die Effizienz dieser (erweiterten) Verfügbarkeit maximierenden Variante zu demonstrieren, wird ein Prototyp unter verschiedenen Testszenarien erprobt. Dabei kommt ein Kriterium zur Konstruktion möglicher Disjunktionen zum Einsatz, das (lokal) die Verfügbarkeit innerhalb von Disjunktion maximiert, indem sich beide Disjunkte einer solchen Disjunktion nur in genau einer Konstante unterscheiden.

Table of Contents

CHAPTER 1

Introduction and Related Work

During the last decades society continuously evolved from an industrial society, in which material goods had the role of a key resource, to a service society built up on information. Hence, one of the key challenges of today's society lies in an effective use and deployment of information. Beside an effective and efficient exchange and processing of information, this key challenge also inevitably comprises effective protection measures to preserve the value of collected information.

1.1 Challenges of Data Publishing

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 lives 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. Hence, a major goal of data publishing lies in the preservation of confidentiality requirements, which is a well-known fundamental protection goal in the field of IT-security [5, 7, 59]. This need for confidentiality preserving data publishing applies in particular to data containing personal information, as surveyed in [56, 92].

Additionally considering the rapid development of computing power, the massive growth of the Internet and the continuous development of information systems [75, 78], collected data is usually managed and shared automatically and the amount of collected data grows steadily. As an immediate consequence of automated publishing of valuable data, the need for an automated and effective enforcement of confidentiality requirements naturally arises to keep control over the publishing of data and to thereby guarantee that each recipient of published data can only retrieve those data records – or, more precisely, the knowledge embodied in these records – he is actually allowed to get to know. Moreover, growing amounts of data naturally require efficient protection mechanisms to prevent an information system from running into an overload when processing huge amounts of data.

At first sight, mechanisms for the enforcement of confidentiality requirements seem to only consider how publishing of data can be blocked effectively. But reconsidering that publishing of appropriate portions of collected data is usually expressly desired, protection mechanisms should (as far as possible) aim at achieving a maximum level of availability and should hence provide as much (knowledge embodied in) data as possible without compromising any confidentiality requirements. Thus, the development of protection mechanisms enforcing confidentiality requirements is usually closely related to finding a reasonable trade-off between confidentiality and availability requirements [5, 7, 59].

1.2 Enforcing Confidentiality

In the course of time, several approaches for the enforcement of confidentiality requirements have been proposed. All of these approaches have in common that they aim at preventively enforcing these confidentiality requirements, as knowledge once, possibly accidentally, revealed to an adversary can – due to the nature of information – *not* be revoked afterwards (cf. [59]). Hence, approaches to detect unwanted disclosures of sensitive information afterwards can only help to mitigate the damage resulting from these disclosures, but can *not* replace mechanisms for the preventive enforcement of confidentiality requirements.

1.2.1 Traditional Access Control

Most approaches for the enforcement of confidentiality requirements operate on the level of raw data and essentially aim at enforcing a collection of access rights specifying which user is (possibly not) allowed to access which data records. Thereby,

these approaches can basically be divided into *discretionary* and *mandatory* access control approaches [5, 7, 44, 59].

Within discretionary approaches each individual user is supposed to be responsible for a certain set of data records and can hence set up access rights for these data records at his own discretion. Such a specification of access rights can thereby range from simple access rules – essentially explicitly listing how to decide on each single request issued by a certain user wishing to access a certain data record – to sophisticated sets of high-level (logic-based) rules, from which explicit access decisions can be derived. Moreover, data records might be structured (as, for example, in relational databases) and different approaches to (discretionary) access control might hence differ in how fine-grained access rights can be specified and in if and to what extent the actual values stored within certain data records can be taken into account for decisions on access requests.

Mandatory approaches instead rely on the assumption that a single security officer mandatorily sets up a system-wide set of access rights. These access rights are specified by considering a partially ordered set of confidentiality levels and by assigning such a confidentiality level to each data record as a classification and to each individual user as a clearance. A user's request to retrieve a data record is then approved, if his clearance is at least as high as the classification of the data record he wishes to read. Otherwise, his request is denied. This way of deciding on access requests obviously aims at controlling data flows – and hence also the flow of the information embodied in this data – by ensuring that data can only flow into the direction of at least equally high confidentiality levels.

Under the supposition that the values of data records can change over the time and that the partially ordered set of confidentiality levels forms a finite lattice, the above sketched mandatory approach can be refined to capture the evolution of the sensitivity of data records and to thereby introduce a kind of "history-awareness". This is essentially achieved by updating the security level of a data record each time a value of this record is changed and this data record is *not* classified as least as high as (the source of) the new data flowing into this record. In this case, the supremum of the considered data record and the new data flowing into this record – whose existence is guaranteed by the properties of a finite lattice – is chosen as the new classification of this record.

1.2.2 The Need for Inference Control

Although approaches to access control have been continuously improved in terms of expressiveness, granularity, content-sensitivity and even history-awareness as

sketched above, they nonetheless operate essentially on the level of raw data. But usually, a data owner's confidentiality requirements primarily aim at keeping certain pieces of knowledge secret. Although this obviously means that those data records containing this confidential knowledge need to be kept secret, there is nonetheless a crucial difference between the protection of raw data and the protection of knowledge: while the protection of raw data can essentially be achieved by preventing the delivery of certain data records to certain users, the protection of knowledge requires to actually *not* enable an adversary to infer confidential pieces of knowledge – even if this adversary tries to logically deduce confidential knowledge on the basis of (the knowledge embodied in) the data revealed to him and by additionally exploiting his general awareness of the protection mechanism and some further a priori knowledge he might have [7, 9, 44].

To exemplify this difference, consider a hospital running an information system in order to store the disease of each of its patients together with the prescribed medication to cure this disease. In terms of privacy concerns, only doctors are supposed to access the disease from which a patient suffers. Nurses are *not* allowed to access a patient's disease, but are instead allowed to access the prescribed medication to enable them to take care of a patient's medical treatment. Within this scenario, a curious nurse might nonetheless be able to infer the disease a certain patient suffers from by simply considering which diseases can be cured by the combination of drugs this patient takes: if there is only one disease cured by this combination of drugs, she can logically deduce that the considered patient suffers from this disease, although she is *not* able to (directly) access the corresponding value of the patient's data record.

Inference control hence means to protect the knowledge embodied in (elements of) a confidentiality policy by suitably confining an adversary's possible gain of information such that this adversary is *not* able to infer a piece of knowledge to be kept confidential by employing his reasoning capabilities [9, 16]. However, this presupposes that data is handled on the level of its semantics and that further a notion of implication (or entailment) between pieces of knowledge is provided on the basis of this semantics [8, 10].

As surveyed in [55], several approaches to inference control have been explored in the course of time, ranging from Denning's seminal work [49] on inference control for statistical databases and information flows in programs to Halpern's and O'Neill's work [61] emphasizing a purely abstract approach to achieve sophisticated information-theoretic confidentiality within a high-level multiagent system framework. On this thesis the seminal work presented in [82] and later revived in [40] has a major influence, which aims at suitably confining the answers an information system replies to a user's queries such that this user is provably *not* able

to compromise any confidentiality requirements – and thereby lays the foundation for the framework of so-called "Controlled Interaction Execution", in which this thesis is placed. Within this framework of Controlled Interaction Execution also an instantiation of Halpern's and O'Neill's approach has been developed in [32] taking a closer look at the connection between these approaches.

1.2.3 Controlled Interaction Execution

As outlined above, inference control can only be implemented successfully, if data is handled on the level of its semantics and a notion of implication (or entailment) is provided on the basis of this semantics. For that reason, the framework of Controlled Interaction Execution is logic-based in the sense that all considered knowledge is supposed to be modeled within some well-defined logic framework, whose semantics is comprehensive enough to capture an adversary's reasoning capabilities [8, 10, 12]. Usually, a fragment of first-order logic [76], which is suitably tailored to the needs of relational databases [68], is employed as such a logic framework (cf. [15, 19, 27, 30, 38]), as it is well understood that first-order logic provides a solid foundation for the modeling of relational databases [1]. But there are also approaches relying on a non-monotonic logic framework (cf. [34]) to model an adversary's reasoning capabilities more appropriately and approaches relying on propositional logic (cf. [36]) to simplify the overall modeling.

Due to the above gained insight that (harmful) inferences can be often drawn by combining several pieces of (per se harmless) knowledge, it is indispensable that the protection mechanism obtains an (as far as possible) complete picture of an adversary's view on a considered original database instance [8, 10, 12]. Thereby, an adversary's view on this original instance is established by his interactions with the (protection mechanism of the) considered information system and some further a priori knowledge he might possibly have independently of his interactions with this information system. Typical examples of such a priori knowledge are database constraints arranged for the database schema underlying the considered original instance, but also some knowledge about the world in general – such as the knowledge which diseases can be cured by which combination of drugs, as assumed in the above given example.

To express the knowledge to be kept confidential from a certain adversary, a declarative confidentiality policy is set up [8, 10, 11]. In most cases, such a confidentiality policy consists of a finite set of so-called potential secrets, each of which is a sentence of the employed logic framework and expresses that the considered adversary must *not* be enabled to get to know that the knowledge embodied in this

potential secret is satisfied by a considered original database instance. So, regardless of whether this potential secret is actually satisfied by this original instance or not, from an adversary's view on the original database instance – established by his interactions with the information system and his a priori knowledge – it must always be possible that this potential secret is *not* satisfied.

Consequently, a confidentiality policy can only be enforced, if for each of its potential secrets the existence of an alternative database instance obeying – i.e., *not* satisfying – this potential secret appears to be possible from an adversary's point of view. But such an alternative instance is *not* credible, if an adversary is able to exclude it from being the "real" original instance by distinguishing it from this original instance. A protection mechanism should hence be designed such that

- for each potential secret of a considered confidentiality policy the existence of an alternative database instance obeying this potential secret is guaranteed such that further
- each view an adversary can possibly gain on the original instance – on the basis of his interactions with the (protection mechanism of the) information system and his further a priori knowledge – remains the same, if the protection mechanism of the information system is instantiated with this alternative instance instead of the original instance.

In this case, the alternative instance obeying the considered potential secret is indistinguishable from the original instance from an adversary's point of view [8, 10, 11], even if the original instance does *not* obey this potential secret and an adversary is further supposed to be aware of the protection mechanism instantiated with all input parameters except for the original database instance to be protected. This assumption that an adversary is supposed to have as much knowledge as possible – and is, in particular, therefore usually also supposed to be aware of the confidentiality policy set up for him – complies with the maxim "no security by obscurity", which is one of the principle design goals of Controlled Interaction Execution [7, 8, 10, 12].

The above given confidentiality requirement, which is also referred to as "inference-proofness", can actually be implemented by dynamic or static protection mechanisms. The dynamic approaches (cf. [13, 15]) aim at controlling an adversary's gain of knowledge at runtime and must hence dynamically decide for each interaction request issued by an adversary, whether the reaction to this request needs to be distorted because of enabling the adversary to compromise the confidentiality policy. As an adversary's view on the original instance hence evolves over the sequence of issued interaction requests, all reactions an adversary receives in response to his requests need (in general) to be recorded to be able to decide

whether the knowledge embodied in these reactions is harmful in combination with reactions to subsequent interaction requests.

In contrast, the static approaches (cf. [27, 30, 38]) first construct an inference-proof materialized view on a given original database instance in a preprocessing step. Although the protection mechanism aims at keeping this view as close to the original instance as possible, it introduces distortions wherever necessary to ensure that the knowledge of this materialized view does *not* enable an adversary to compromise the confidentiality policy set up for him. After this preprocessing step, an adversary's interaction requests can then be processed safely on the basis of this materialized view without any further interception.

Until now, two different techniques to distort confidentiality compromising knowledge have been explored [8, 10, 11]. One is the explicit refusal of such knowledge in the sense that an adversary is explicitly notified which pieces of knowledge are *not* revealed to him. Employing this technique of course introduces the challenge that protection mechanisms should be well prepared for adversaries trying to take advantage of this additional knowledge in the form of refusal notifications. The other distortion technique is to introduce lies in the sense that non-valid knowledge might be declared as valid knowledge and vice versa. This technique introduces the challenge of keeping an adversary's view on the original database instance consistent in order to prevent this adversary from inferring which truth values are modified and thereby compromising the confidentiality policy. As demonstrated in [14], a combined usage of both of these distortion techniques is possible, too.

1.3 Design Goals and Scope

Reconsidering the challenges of data publishing discussed in Section 1.1, the goal of this thesis is to develop a novel approach allowing for confidentiality preserving data publishing within the framework of Controlled Interaction Execution. This approach to be developed should thus allow for the construction of an inference-proof materialized view of a given complete database instance, which does provably *not* enable a considered adversary to compromise a single element of a confidentiality policy and can hence be safely released to this adversary.

1.3.1 Requirement Analysis

Following the notion of inference-proofness provided by the framework of Controlled Interaction Execution, the approach to be developed should construct materialized views satisfying the requirement that for each element of a given

confidentiality policy the existence of a complete alternative database instance is guaranteed such that

- this alternative instance obeys the considered policy element,
- this alternative instance nonetheless satisfies an adversary's a priori knowledge, including semantic database constraints an adversary is supposed to be aware of, and
- the secure materialized view the protection mechanism would construct for this alternative instance is – from an adversary's point of view – indistinguishable from the actually released secure materialized view the protection mechanism returned for the given original database instance.

In fact, an approach generating inference-proof materialized views within the framework of Controlled Interaction Execution has already been proposed in [36, 37], then refined in [38, 91] and finally been implemented as described in [18]. This approach essentially aims at achieving inference-proofness by modifying the truth-values of some database tuples in the sense that valid tuples of a given complete original database instance may become non-valid and non-valid tuples – i.e., tuples, which are *not* contained in a given complete original database instance – may instead become valid, thereby inducing a complete alternative database instance. To keep availability as high as possible, the number of these modified truth-values is minimized, i.e., it is just high enough to provably achieve inference-proofness.

Releasing such an alternative instance with modified truth values instead of the original instance clearly means that a user's view on the original database instance contains lies, i.e., knowledge *not* complying with the real-world scenario captured by the considered original database instance. To further prevent an adversary from reconstructing original truth-values, it is of crucial importance that an adversary is *not* able to detect which of the truth values are actually modified. Hence, there might be some ethical concerns to release such an alternative instance and there might even be some application scenarios, in which these distortions in the form of lies mislead a legal user accidentally considering some censored knowledge to draw wrong – and possibly even dangerous – conclusions on the basis of the data released to him. Moreover, this approach also suffers from its high computational complexity, as experimentally evaluated in [18].

Accordingly, the approach to be developed should generate secure materialized views, which – despite distortions, possibly introduced by the protection mechanism – contain only true knowledge, which is *not* in conflict with the original database instance. Additionally, it seems worthwhile that these distortions should be readily identifiable for anyone using the released secure view. So, similar to those approaches to Controlled Interaction Execution, which dynamically respond

to interaction requests and, if necessary, explicitly refuse harmful requests by returning a distinguished refusal notification [8, 10, 11, 12, 13], a distinguishing feature making distortions identifiable and further providing a basis for achieving inference-proofness should be selected as a means to distort knowledge.

In fact, one could even achieve this goal by employing an approach developed in [15], which dynamically responds to a series of interaction requests in the form of domain-independent open queries of a decidable subclass of first-order logic, i.e., evaluable queries containing free variables. For each of these open queries this approach returns a response, which answers for each of the (infinitely many) constant substitutions of the free variables of this open query whether this query is satisfied by the given complete original database instance under the considered constant substitution of the free variables or not, provided that this answer for the particular constant substitution – together with previous answers and a priori knowledge – does *not* enable an adversary to compromise a considered confidentiality policy. Otherwise, this answer is distorted by returning a distinguished refusal notification.

This approach, which essentially answers an open query by simulating it by an appropriate (finite) series of closed queries, can also handle an open query asking for the whole database relation, i.e., an open query over a considered database relation containing only free variables and *no* constant symbols. Due to the inference-proofness of the approach (cf. [15]) the answer to such a query immediately corresponds to an inference-proof materialized view. But, as experimentally evaluated in [17], this approach suffers from its high computational complexity and is hence only applicable for small input instances.

Accordingly, the approach to be developed should be efficient enough to handle even large input instances resulting from steadily growing collections of data (cf. Section 1.1) within a reasonable timeframe. As inference-proofness follows the goal that an adversary should *not* be able to (logically) infer pieces of knowledge protected by a confidentiality policy, it generally requires costly theorem proving on the level of an employed logic framework to check for harmful implications [8, 10, 11, 12]. To be able to achieve inference-proofness efficiently, the approach to be developed should rely on a restricted fragment of a suitable logic framework, within which the validity of implication relationships can be decided efficiently without costly general theorem proving [69].

Such approaches for the construction of materialized views achieving confidentiality requirements without general theorem proving are proposed in [45, 46] and in [2, 57]. Both of these approaches aim at achieving confidentiality by fragmenting a given database instance vertically into (at least) two fragments, each of which consists of a subset of columns of the original database instance. Thereby,

the approach proposed in [45, 46] aims at the construction of one non-secure fragment, which must be kept secret, and one secure fragment to be released to an adversary. The approach proposed in [2, 57] instead aims at the construction of a fragmentation such that exactly one of its fragments can be chosen freely to be released to an adversary, but resorts to encryption of certain columns whenever the enforcement of all confidentiality requirements is *not* possible on the sole basis of fragmentation. From an adversary's point of view the original database instance is then split into one part of visible (cleartext) columns and one refused part consisting of all non-visible (and all encrypted) columns. As he is supposed to know the database schema underlying the original instance, an adversary can readily identify which columns are refused, but he does *not* get to know their values.

As shown in [27] and [30], both of these approaches are able to achieve inference-proofness in the sense of Controlled Interaction Execution, provided that an adversary's a priori knowledge and the considered confidentiality requirements "fit together" structurally. In terms of efficiency, these fragmentation approaches have the advantage that the computationally expensive task of determining a confidentiality preserving fragmentation is solely performed on the level of the typically small database schema, while the construction of fragments on the instance level essentially corresponds to the computationally inexpensive task of constructing projections of the original database instance on the schemas of the fragments.

Although these fragmentation approaches comply with the requirements set up for the approach to be developed so far, they suffer from the major drawback that confidentiality requirements can only be specified in a coarse-grained way on the level of the database schema – in the sense that associations between (the values of) certain columns should *not* be revealed. But in terms of availability a more fine-grained specification of confidentiality requirements in the spirit of potential secrets (cf. Section 1.2.3) is desirable. Moreover, even if the sentences of an adversary's a priori knowledge stem from the considered subclass of sentences, the existence of an inference-proof fragmentation is *not* always guaranteed, as the structure of an adversary's a priori knowledge and the structure of the considered confidentiality requirements need to "fit together" (cf. [27]).

Another already existing approach introduced in [4] aims at preserving confidentiality by replacing certain values of certain database tuples of a given original instance by `null`-values. Thereby, this approach relies on the assumptions that the given original instance is possibly incomplete in the sense that it may already contain `null`-values and that an adversary does *neither* know which knowledge is to be kept confidential *nor* with which semantic constraints the original database instance has to comply according to its underlying database schema. This approach claims that hence confidentiality is achieved, as an adversary is – in contrast to the

above mentioned design goal that distortions should be readily identifiable – *not* able to distinguish between those `null`-values already contained in the original instance and those `null`-values additionally introduced to suppress confidential knowledge. While the former assumptions of incomplete databases [35, 8, 10] and unknown confidentiality policies [6, 8, 10, 11] still comply with at least some approaches to Controlled Interaction Execution, the latter assumption of unknown (but existing) semantic database constraints is in direct conflict with the principle design goals of Controlled Interaction Execution, always following the maxim "no security by obscurity" (cf. Section 1.2.3).

All of the above discussed existing approaches enforce confidentiality requirements by either deliberately introducing wrong knowledge or by refusing certain pieces of knowledge. Although the above mentioned design goals that only true knowledge should be revealed and that distortions should be readily identifiable seem to immediately suggest to employ explicit refusals within the approach to be developed, the well-known approaches of k-anonymization and ℓ-diversification [47, 70, 79, 86] propose a less severe means: these approaches aim at preventing the re-identification of individuals on the basis of so-called quasi-identifiers, which describe some of the individuals' properties, by generalizing the values of these quasi-identifiers to wider sets of possible values.

Within the logic-based framework of Controlled Interaction Execution one could analogously try to adapt this generalization of knowledge by replacing confidential (definite) knowledge by weaker disjunctive knowledge. These disjunctions should contain only true knowledge, but limit an adversary's possible gain of information to such an extent that this adversary is provably *not* able to compromise an element of a considered confidentiality policy. Compared to the complete refusal of confidential knowledge – which corresponds to its maximum generalization – this weakening of knowledge is more cooperative in terms of availability in the sense that more information about the original database instance is revealed. Moreover, an adversary is easily able to identify this weakened knowledge due to its distinguished (syntactic) representation.

As a consequence of this additional knowledge provided by disjunctions, particular attention must be paid to eliminate so-called *meta-inferences* [8, 10, 24]. A piece of confidential knowledge is inferred with the help of such a meta-inference, if it is obtained by excluding all possible alternative settings, under which this knowledge is *not* valid, by simulating these alternative settings as inputs for the algorithm generating the (inference-proof) materialized views and by being then able to distinguish the materialized views resulting from each of the alternative settings from the released one. Then, an adversary is able to conclude that only settings,

under which this specific piece of knowledge is true, can comply with the original database instance of his interest.

1.3.2 Survey of the Weakening Approach

This thesis introduces a novel approach within the framework of Controlled Interaction Execution creating *inference-proof materialized views* on complete relational database instances, which are suitable for confidentiality preserving data publishing in the sense that they provably comply with a confidentiality policy consisting of potential secrets. To achieve this goal, confidentiality compromising database tuples are replaced by weaker knowledge in the form of disjunctions (preferably) consisting of certain elements of the confidentiality policy. These disjunctions contain only true knowledge, but weaken an adversary's possible gain of information such that this adversary is provably *not* able to infer confidential knowledge.

At first, a *generic* weakening approach is developed, which allows for the construction of non-trivial disjunctions of any length ≥ 2 to weaken an adversary's possible gain of confidential knowledge. Thereby, the achieved level of confidentiality varies with the length of disjunctions, as longer disjunctions of potential secrets obviously provide more alternatives which (combination of) policy elements of such a disjunction might be satisfied by a considered original database instance. Similar to the approaches of k-anonymization [47, 79, 86], which motivated the usage of weakening disjunctions in Section 1.3.1 and aim at generalizing so-called quasi-identifiers to such an extent that an individual can *not* be distinguished from (at least) $k-1$ other individuals on the basis of these quasi-identifiers, a disjunction of length k should *not* enable an adversary to distinguish whether a certain potential secret of this disjunction or one of the $k-1$ other potential secrets of this disjunction is satisfied by the original instance.

Considering confidentiality policies of a non-trivial size naturally raises the question which policy elements should be grouped together to a weakening disjunction: in terms of confidentiality all alternatives provided by a weakening disjunction should be equally probable to prevent an adversary from excluding certain alternatives from being true and in terms of availability such a disjunction should provide as much useful information as possible. Hence, a criterion – which is referred to as a *notion of admissible indistinguishabilities* – specifying which policy elements might be possibly grouped together to an admissible weakening disjunction needs to be specified against the background of a considered application scenario. Similar to clustering techniques known from machine learning (cf. [51, 74, 81]),

such a criterion thereby specifies a kind of "nearness" between elements of a confidentiality policy and thereby allows for a clustering of policy elements to both confidentiality and availability preserving disjunctions.

Although the generic weakening approach specifies such a clustering of policy elements on the declarative level, it does *not* provide an algorithmic instantiation on the operational level. But such an algorithmic instantiation of the clustering of policy elements is then provided for an *availability-maximizing* instantiation of the generic weakening approach, which aims at the construction of availability-maximizing disjunctions of length 2. This clustering is based on well-known and efficient algorithms for the computation of maximum matchings [50, 67, 77, 80] on graphs modeling each potential secret to be clustered as a vertex and each admissible disjunction of length 2 as an edge.

In the field of privacy preserving data publishing there are also other approaches enforcing certain confidentiality requirements on the basis of clusterings on graphs (cf. [43, 62, 56, 92]). But these approaches usually aim at preventing the structural re-identification of the graph itself by making the vertices of each cluster indistinguishable from each other, while the weakening approach developed in this thesis employs the graph as a means to determine clusters, whose vertices in the form of policy elements are to be made indistinguishable from each other regarding their validity status. The graph itself is instead *not* sensitive and an adversary is even supposed to be able to construct this graph himself.

As studied extensively in prior work on Controlled Interaction Execution, an adversary's possibilities to infer confidential knowledge might grow significantly, if he is supposed to have some a priori knowledge [13, 15, 27, 38]. To also explore the impact of an adversary's a priori knowledge on his possibilities to compromise the confidentiality policy within the developed weakening approach, a suitably restricted subclass of so-called "Tuple Generating Dependencies", which are well-known in the field of relational databases [1, 54], is considered. This finally leads to an extended weakening approach, which provably preserves inference-proofness under the considered subclass of a priori knowledge, but still remains computationally efficient even for large input instances.

To be able to fully implement the (extended) availability-maximizing instantiation of the weakening approach to demonstrate its high efficiency experimentally, an example of a notion of admissible indistinguishabilities called *interchangeability* is provided and evaluated. Interchangeability restricts distortion within a disjunction only locally in the sense that a pair of (possibly additional) potential secrets only differs in exactly one constant parameter and thereby generalizes this constant parameter to a wider set of possible values.

1.4 Contributions to Joint Work

The ideas developed in this thesis are extensions of the seminal ideas published in [28] as joint work with my supervisor Joachim Biskup. His contributions to this work comprised joint exploration of potential approaches, ongoing discussions, proof-reading and general advisory. But nevertheless, the proposed seminal ideas to enforce confidentiality policies with the help of weakening disjunctions and their further development to an algorithmic approach, on which this thesis is based, are my own original work.

This thesis significantly extends the seminal ideas published in [28] by essentially (but not limited to)

- giving a complete formalization of the generic weakening approach sketched in [28] and analyzing its complexity,
- providing a detailed exploration of how confidentiality policies with existential quantification can be enforced (which is only sketched briefly in [28]),
- investigating more deeply which requirements criteria specifying the admissibility of weakening disjunctions should fulfill,
- analyzing and disabling harmful inference-channels arising from an adversary's a priori knowledge in the form of a restricted subclass of well-known Tuple Generating Dependencies,
- experimentally evaluating these extensions on the basis of a prototype implementation and
- giving detailed proofs that declarative confidentiality requirements in the sense of Controlled Interaction Execution are met.

Some of these extensions are also sketched briefly in [29], which is accepted for publication as joint work with my supervisor Joachim Biskup and evolved on the basis of the manuscript of this thesis. Thereby, Joachim Biskup contributed to this work by summarizing the above mentioned extensions on an abstract level and discussing improvements of these extensions. But nevertheless, all of these extensions are my own original work.

CHAPTER 2

Inference-Proofness by Weakening

After motivating the development of a novel approach for the construction of inference-proof materialized views on relational database instances, which aims at weakening confidential knowledge with the help of disjunctions, a suitable logic framework needs to be developed to provide a basis for the construction of such a weakening approach within the logic-based framework of Controlled Interaction Execution. To this end, a restricted but expressive subclass of first-order logic is now proposed, which allows for efficient decisions on the validity of implication relationships without costly general theorem proving. Subsequently, the seminal ideas how confidentiality requirements can be enforced within this logic framework on the basis of weakening disjunctions are introduced.

2.1 The Formal Framework

Similar to other approaches to Controlled Interaction Execution, the new approach to be developed is located in the field of relational databases. Thereby, for simplicity, all data is supposed to be represented within a single complete relational database instance r over a database schema $\langle R|\mathcal{A}_R|SC_R\rangle$ with relational symbol R and the finite set $\mathcal{A}_R = \{A_1, \ldots, A_n\}$ of attributes. Moreover, all attributes

are assumed to have the same *fixed but infinite* domain *Dom* of constant symbols (cf. [15, 68]) and the set SC_R is supposed to contain some database constraints (cf. [1]), which must be satisfied by each database instance r constructed over this schema. The further assumption that each database instance r represents *complete information* thereby means that r is supposed to contain a finite set of valid database 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 (cf. [68]).

In compliance with the general framework of Controlled Interaction Execution (cf. Section 1.2.3), a database instance is supposed to be 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 distinguished binary predicate symbol $\equiv$ for expressing equality is set up. To be able to actually model each possible database instance r within this first-order language $\mathscr{L}$, the fixed but infinite domain *Dom*, over which the database tuples of an original instance r are supposed to be constructed, is also taken as the set of constant symbols of the first-order language $\mathscr{L}$. To be further able to also model more complex sentences with variables within this language, an infinite set *Var* of variables is supposed to be available.

All sentences of $\mathscr{L}$ are constructed inductively in the natural fashion [68, 76] using the quantifiers $\forall$ and $\exists$ and the connectives $\neg$, $\wedge$, $\vee$ and $\Rightarrow$. Thereby, each term is either a constant symbol of *Dom* or a variable of *Var* (functions are not allowed) and each variable is supposed to be either universally or existentially quantified. Hence, only closed formulas (i.e., sentences) are supposed to be in $\mathscr{L}$.

This syntactic specification of the first-order language $\mathscr{L}$, which is obviously tailored to the specific needs of the logic-oriented modeling of relational database instances, is now also complemented with a semantics reflecting the characteristics of databases [15, 68]. Such a database-specific semantics essentially restricts commonly known semantics for first-order logic (cf. [76, 88]) in the sense that

- *no* further semantics is provided for the constant symbols of the first-order language $\mathscr{L}$ and hence each constant symbol of $\mathscr{L}$ is interpreted by itself and is moreover seen to be equal only to itself and
- interpretations of the predicate symbol R, which represent a finite Herbrand interpretation of (a logic-oriented modeling of) a database instance r consisting of only a finite number of database tuples, can correspondingly only be of finite size.

This kind of database-specific semantics for first-order logic is also referred to as a DB-Interpretation and can be formally captured as follows.

Definition 2.1: DB-Interpretation

Given the first-order language $\mathscr{L}$ with the set *Dom* of constant symbols, an interpretation $\mathcal{I}$ is a *DB-Interpretation* for $\mathscr{L}$, if

(i) the set *Dom* of constant symbols is employed as the universe of $\mathcal{I}$,

(ii) each element (constant symbol) of the universe of $\mathcal{I}$ is interpreted by itself, i.e., $\mathcal{I}(v) = v$ holds for each $v \in Dom$,

(iii) the predicate symbol R for modeling database tuples of arity n is interpreted by a finite relation $\mathcal{I}(R) \subset Dom^n$ and

(iv) the distinguished binary predicate symbol $\equiv$ is interpreted by the fixed and infinite relation $\mathcal{I}(\equiv) = \{\, (v, v) \mid v \in Dom \,\}$.

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* of sentences of the constructed first-order language $\mathscr{L}$ by a DB-Interpretation is the same as in usual first-order logic, i.e., whether a sentence $\Phi \in \mathscr{L}$ is satisfied by a DB-Interpretation $\mathcal{I}$ or not is evaluated in the natural fashion over the inductive structure of this sentence Φ [76, 88]. Similarly, the notion of *implication* (or entailment) under DB-Semantics corresponds to the notion of implication under usual first-order semantics.

Definition 2.2: DB-Implication

Consider the constructed first-order language $\mathscr{L}$. A set $\mathcal{S}$ of sentences of $\mathscr{L}$ *implies* a sentence Φ of $\mathscr{L}$ under DB-Implication, i.e.,

$$\mathcal{S} \models_{DB} \Phi \ ,$$

if and only if each DB-Interpretation $\mathcal{I}$ satisfying $\mathcal{S}$ (written as $\mathcal{I} \models_M \mathcal{S}$) also satisfies Φ (written as $\mathcal{I} \models_M \Phi$).

Slightly abusing the notation presented in this definition, DB-Implication under a singleton set $\mathcal{S} = \{\Psi\}$ is in the following usually written as $\Psi \models_{DB} \Phi$. Moreover, for convenience, DB-Implication is in the following often simply referred to as "implication", when it is clear from the context that DB-Implication is meant.

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

(a) Complete instance r

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

(b) Logic-oriented modeling of r

Figure 2.1: Logic-oriented modeling of a complete database instance

As required by the framework of Controlled Interaction Execution, the constructed first-order language $\mathscr{L}$ can be employed to model complete database instances logic-orientedly. This is exemplified in Figure 2.1, in which a complete database instance r is modeled as a set of first-order sentences of $\mathscr{L}$. Thereby, each valid tuple $\boldsymbol{c} \in r$ is modeled as a corresponding ground atom $R(\boldsymbol{c})$ of $\mathscr{L}$ and the infinite set of invalid tuples – which is not explicitly enumerable – is expressed implicitly with the help of a so-called completeness sentence (cf. [15]) of the form

$$(\forall X_1) \ldots (\forall X_n) \left[\bigvee_{\nu \in r} \left(\bigwedge_{A_j \in \mathcal{A}_R} (X_j \equiv \nu[A_j]) \right) \vee \neg R(X_1, \ldots, X_n) \right]$$

having a universally quantified variable X_j for each attribute $A_j \in \mathcal{A}_R$ of the corresponding database schema. This completeness sentence expresses that each constant combination $(c_1, \ldots, c_n) \in Dom^n$, with which the universally quantified variables $X_1, \ldots, X_n$ can actually be substituted, is either explicitly excluded from being invalid because of the disjunct

$$X_1 \equiv c_1 \wedge \ldots \wedge X_n \equiv c_n$$

or is invalid, as the sentence $\neg R(c_1, \ldots, c_n)$ holds. By construction, this completeness sentence is satisfied by the DB-Interpretation $\mathcal{I}_r$ induced by r.

In general, implication between (sets of) sentences of first-order logic is known to be computationally undecidable [42]. Although there are some rather expressive classes of first-order logic, which allow for algorithmic theorem proving [15, 42], implication is in general still computationally hard to decide within these classes. To be able to nonetheless decide on the validity of implication relationships between first-order sentences efficiently, one must hence restrict to less expressive subclasses of first-order logic [69]. Such a subclass of first-order logic is known from the approaches proposed in [19, 22, 23, 24, 25, 26, 69] and is in the following referred to as existentially quantified atoms.

Definition 2.3: Existentially Quantified Atom

A sentence of the first-order language $\mathscr{L}$ 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 occurs only once, i.e., $t_i \neq t_j$ for all $t_i, t_j \in \boldsymbol{X}$ with $i \neq j$.

In particular, ground atoms of the form $R(c_1, \ldots, c_n)$ with each term c_i being a constant symbol of *Dom* are also considered to be (a restricted kind of) existentially quantified atoms with an empty set of existentially quantified variables. For this special kind of existentially quantified atoms the implication problem is obviously easy to solve: one ground atom implies another ground atom, *if and only if* both of these ground atoms are equal.

But also for those existentially quantified atoms actually containing variables the implication problem under DB-Semantics can be reduced to a simple pattern matching problem, as there can *not* be any implicit equalities between different terms of an existentially quantified atom due to the syntactic restriction that each variable can occur only once within each existentially quantified atom. This reduction of the implication problem under DB-Semantics to an efficiently decidable pattern matching problem is formally captured in the following.

Lemma 2.1: DB-Implication for Existentially Quantified Atoms

Suppose that both $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ are existentially quantified atoms. The DB-Implication

$$(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n) \models_{DB} (\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$$

holds, *if and only if* 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 $t_i = \bar{t}_i$.

Proof. To start with the *only-if-part*, consider the given existentially quantified atoms $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ and suppose that 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 $t_i = \bar{t}_i$. Moreover, consider an arbitrary DB-Interpretation $\mathcal{I}$ satisfying $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$. As this sentence is obviously satisfiable and *not*

tautological, this DB-Interpretation $\mathcal{I}$ must contain a tuple $(c_1, \ldots, c_n) \in \mathcal{I}(R)$ with $c_i = t_i$ for each $i \in \{1, \ldots, n\}$ with $t_i \in \mathit{Dom}$. But such a DB-Interpretation $\mathcal{I}$ also satisfies $(\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ due to $\bar{t}_i = t_i = c_i$ for each $i \in \{1, \ldots, n\}$ with $\bar{t}_i \in \mathit{Dom}$ because of the assumption that 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 $t_i = \bar{t}_i$ and because of $c_i = t_i$ for each $i \in \{1, \ldots, n\}$ with $t_i \in \mathit{Dom}$. Hence, each DB-Interpretation satisfying $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ also satisfies $(\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ and the validity of the DB-Implication $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n) \models_{DB} (\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ is a direct consequence.

To now prove the *if-part* by contraposition, again consider the given existentially quantified atoms $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ and suppose that there is an $m \in \{1, \ldots, n\}$ such that $\bar{t}_m$ is a constant symbol of Dom and t_m is either a variable of $\boldsymbol{X}$ or a constant symbol of Dom with $t_m \neq \bar{t}_m$. Next, consider a DB-Interpretation $\mathcal{I}$ with $\mathcal{I}(R) = \{\, (c_1, \ldots, c_n) \,\}$ and with further

- $c_i = t_i$ for each $i \in \{1, \ldots, n\} \setminus \{m\}$ with $t_i \in \mathit{Dom}$
- $c_m = t_m$, if t_m is a constant symbol of Dom and
- $c_m \in \mathit{Dom} \setminus \{\bar{t}_m\}$, if t_m is a variable of $\boldsymbol{X}$.

This DB-Interpretation $\mathcal{I}$ obviously satisfies $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$, but it does *not* satisfy $(\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ because of $\bar{t}_m \neq c_m$ due to either $c_m = t_m \neq \bar{t}_m$ or $c_m \in \mathit{Dom} \setminus \{\bar{t}_m\}$ and as a direct consequence the given implication relationship $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n) \models_{DB} (\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ does *not* hold, either. ♠

To actually benefit from these efficiently decidable implication relationships within the framework of Controlled Interaction Execution, the confidentiality policies employed within this framework – for whose elements (expressing user-specific prohibitions) the non-implication must be guaranteed to achieve the wanted confidentiality requirements [11] – should reasonably be restricted to the above mentioned subclass of existentially quantified atoms. This leads to the following (restricted) definition of confidentiality policies consisting of potential secrets.

Definition 2.4: Confidentiality Policy

A *potential secret* Ψ is an existentially quantified atom of the first-order language $\mathscr{L}$ and a *confidentiality policy psec* is a finite set of potential secrets.

As usual, the semantics of a potential secret Ψ requires that an adversary must *not* be able to infer that the information embodied in Ψ is satisfied by a considered

(original) database instance (cf. Section 1.2.3). So, regardless of whether Ψ is actually satisfied by this database instance or not, from the adversary's point of view – established by a possibly weakened view on the original instance – it must always be possible that Ψ is *not* true. Correspondingly, a (possibly alternative) database instance r is supposed to *obey* a potential secret Ψ of a confidentiality policy *psec*, if r does *not* satisfy Ψ, i.e., $\mathcal{I}_r \not\models_M \Psi$. Moreover, this instance r *obeys* the confidentiality policy *psec*, if r obeys each potential secret of this policy.

2.2 Basic Ideas of Protecting Information

To now start the development of an algorithmic approach generating inference-proof materialized views within the framework of Controlled Interaction Execution, for now consider a *simplified* setup of input instances consisting of

- a *simple* confidentiality policy $psec = \{\Psi_1, \Psi_2\}$ containing exactly two potential secrets in the form of ground atoms and
- a complete original database instance r over a database schema $\langle R|\mathcal{A}_R|\,\emptyset\,\rangle$ without any database constraints.

Moreover, suppose that an adversary does *not* have any further a priori knowledge about the original instance r, either.

2.2.1 A First Simple Weakening Approach

Summarizing the requirements collected in Section 1.3, the approach to be developed should enforce a given confidentiality policy *psec* by replacing any knowledge about an original instance r, which might enable an adversary to compromise this confidentiality policy, by weaker knowledge in the form of suitable disjunctions. Thereby, these distortions are readily identifiable for a user due to their distinguished syntactic form. This results in a (possibly incomplete) weakened view $weak(r, psec)$ on the original instance r, which should further

- contain only true knowledge complying with the original database instance r, i.e., $\mathcal{I}_r \models_M weak(r, psec)$, and
- be inference-proof in the sense that for each potential secret $\Psi \in psec$ the existence of a complete alternative database instance r^Ψ over the schema $\langle R|\mathcal{A}_R|\,\emptyset\,\rangle$ is guaranteed such that
  - this alternative instance r^Ψ obeys the considered potential secret Ψ, i.e., $\mathcal{I}_{r^\Psi} \not\models_M \Psi$, and

– the weakened view $weak(r^{\Psi}, psec)$ the algorithm would compute to enforce $psec$ under the considered alternative instance r^{Ψ} is (from an adversary's point of view) indistinguishable from the weakened view $weak(r, psec)$ the algorithm has actually computed to enforce $psec$ under the original instance r, i.e., $weak(r^{\Psi}, psec) = weak(r, psec)$.

Under the *simplified* setup – assuming that a simple confidentiality policy $psec = \{\Psi_1, \Psi_2\}$ of only ground atoms and an original database instance r without any semantic database constraints are given – an algorithm computing such a weakened view $weak(r, psec)$ can be easily sketched as follows [28]:

- for each tuple $\boldsymbol{c} \in r$ with both $R(\boldsymbol{c}) \neq \Psi_1$ as well as $R(\boldsymbol{c}) \neq \Psi_2$ the corresponding ground atom $R(\boldsymbol{c})$ is added to the (initially empty) weakened view $weak(r, psec)$;
- if at least one of the ground atoms of the given confidentiality policy $psec$ is satisfied by the original instance r, i.e., $\mathcal{I}_r \models_M \Psi_1 \vee \Psi_2$, the (weaker) *disjunctive knowledge* $\Psi_1 \vee \Psi_2$ is added to the weakened view $weak(r, psec)$ instead of the definite knowledge that Ψ_1 or (and, respectively) Ψ_2 is valid;
- finally, a (partial) completeness sentence exposing the non-validity of all other constant combinations, which *neither* correspond to a ground atom *nor* to a disjunct of the (so far constructed) weakened view $weak(r, psec)$, is added to $weak(r, psec)$.

Thereby, in contrast to the original database instance r, a total order is supposed to be defined on the sentences that (might) occur in a weakened view on this instance r (cf. [15]). This guarantees that an alternative instance r^{Ψ} with $\mathcal{I}_{r^{\Psi}} \models_M weak(r, psec)$ is *not* distinguishable from a considered original instance r on the basis of a different (syntactic) arrangement of the sentences of its weakened view $weak(r^{\Psi}, psec)$ compared to the weakened view $weak(r, psec)$ released for r. Otherwise, an adversary might simulate the (deterministic) weakening algorithm – he is supposed to be aware of – for this alternative instance r^{Ψ} and might then be able to draw the meta-inference (cf. Section 1.3) that r^{Ψ} can *not* be the original instance leading to the released weakened view $weak(r, psec)$ because of $weak(r^{\Psi}, psec) \neq weak(r, psec)$ due to a different (syntactic) presentation. In this case the adversary would hence be able to exclude r^{Ψ} from being the "real" original instance of his interest, thereby making this alternative instance useless.

To exemplify the importance of this kind of normalization, assume that a weakening algorithm is constructed such that those disjuncts of a weakening disjunction, which are actually satisfied by a considered original database instance, are printed first. Considering a weakening disjunction $\Psi_1 \vee \Psi_2$ of a released weakened view, an

adversary can then easily reason that Ψ_1 must be satisfied by the original instance – by simply considering the order of the elements within this disjunction – and can thereby compromise the confidentiality policy containing Ψ_1 as a potential secret. Further, each alternative instance obeying Ψ_1 – and hence satisfying Ψ_2 instead of Ψ_1 – can easily be distinguished from the original instance satisfying Ψ_1, as the weakening algorithm would return the weakening disjunction $\Psi_2 \vee \Psi_1$ for this alternative instance, which syntactically differs from the disjunction $\Psi_1 \vee \Psi_2$ of the released weakened view on the considered original instance.

As discussed above, a completeness sentence of the logic-oriented modeling of a complete database instance indeed guarantees that this logic-oriented modeling provides *complete knowledge* in the sense that there is exactly one DB-Interpretation satisfying this logic-oriented modeling. Hence, for each possible constant combination $\boldsymbol{c} \in \mathit{Dom}$ over the attributes of the corresponding database schema either the ground atom $R(\boldsymbol{c})$ or its negation $\neg R(\boldsymbol{c})$ is implied by this logic-oriented modeling of the considered database instance.

In contrast, a completeness sentence used within a weakened view on a complete database instance may reveal only *partial knowledge*. In case that this weakened view contains the disjunction $\Psi_1 \vee \Psi_2$ instead of the definite knowledge about which of the potential secrets Ψ_1 and Ψ_2 are actually satisfied by the considered complete database instance, the (partial) completeness sentence does *not* reveal any knowledge about a possible non-satisfaction of any of these potential secrets by its construction – even if one of these potential secrets is actually *not* satisfied by the considered database instance. Correspondingly, there is more than one DB-Interpretation satisfying this weakened view and among these DB-Interpretations there is one that satisfies Ψ_1 and does *not* satisfy Ψ_2 and another one that satisfies Ψ_2 and does *not* satisfy Ψ_1. Hence, for each of the sentences Ψ_1 and Ψ_2 *neither* its validity *nor* its non-validity can be deduced. Of course, any knowledge about the (non-)validity of any of these policy elements is deliberately *not* revealed by the weakened view to enforce the confidentiality policy. Hence, a weakened view may remain incomplete by design.

Nonetheless, the algorithm sketched above immediately satisfies the above mentioned goal that each weakened view $\mathit{weak}(r, \mathit{psec})$ on an original instance r should contain only true knowledge complying with this instance r. The disjunction $\Psi_1 \vee \Psi_2$ weakening the knowledge about which of the potential secrets Ψ_1 and Ψ_2 are satisfied by the original instance r is only added to $\mathit{weak}(r, \mathit{psec})$, if at least one of these potential secrets is actually satisfied by r. If instead both Ψ_1 and Ψ_2 are *not* satisfied by r, the completeness sentence of $\mathit{weak}(r, \mathit{psec})$ reveals this definite knowledge about the non-satisfaction of these policy elements.

Moreover, each weakened view $weak(r, psec)$ returned by this algorithm is also inference-proof: if the disjunction $\Psi_1 \vee \Psi_2$ is contained in $weak(r, psec)$ and both $\Psi_1 = R(\boldsymbol{c}_1)$ and $\Psi_2 = R(\boldsymbol{c}_2)$ with $\boldsymbol{c}_1, \boldsymbol{c}_2 \in Dom^n$ is supposed to hold,

- a complete alternative database instance r^{Ψ_1} obeying the potential secret $\Psi_1 \in psec$ can be constructed by $r^{\Psi_1} := (r \setminus \{\boldsymbol{c}_1\}) \cup \{\boldsymbol{c}_2\}$ and

- a complete alternative database instance r^{Ψ_2} obeying the potential secret $\Psi_2 \in psec$ can be constructed by $r^{\Psi_2} := (r \setminus \{\boldsymbol{c}_2\}) \cup \{\boldsymbol{c}_1\}$.

In both of these sub-cases the resulting alternative instances r^{Ψ_1} and r^{Ψ_2} obviously satisfy the disjunction $\Psi_1 \vee \Psi_2$, just as the original instance r. Accordingly, if an adversary simulates the weakening algorithm on these alternative instances, both of the corresponding weakened views $weak(r^{\Psi_1}, psec)$ and $weak(r^{\Psi_2}, psec)$ are indistinguishable from the weakened view $weak(r, psec)$ released for the original instance r, provided that the sentences of these sequences are arranged in the same order (to mitigate meta-inferences as argued above). So, from this adversary's point of view, the considered instances r, r^{Ψ_1} and r^{Ψ_2} are pairwise indistinguishable, too, because of $weak(r, psec) = weak(r^{\Psi_1}, psec) = weak(r^{\Psi_2}, psec)$.

If the disjunction $\Psi_1 \vee \Psi_2$ is instead *not* contained in $weak(r, psec)$, the original instance r already obeys both of the potential secrets Ψ_1 and Ψ_2 and hence – according to the semantics of potential secrets – *no* distortions are necessary to enforce the confidentiality policy. As a consequence, alternative instances obeying the potential secrets Ψ_1 and Ψ_2 – as required by the definition of inference-proofness – can easily be constructed by $r^{\Psi_1} := r$ and $r^{\Psi_2} := r$, thereby obviously both obeying Ψ_1 and Ψ_2 and also satisfying the property of indistinguishability.

At first sight, one might argue that this weakening algorithm decreases availability more than necessary, as it always weakens the knowledge about the validity of both policy elements Ψ_1 and Ψ_2, even if only one of these policy elements is actually satisfied by the given original database instance. But under the supposition that only knowledge about those potential secrets, which are actually satisfied by a given original instance, is weakened, such a modified algorithm would *not* guarantee the existence of indistinguishable alternative instances as required by inference-proofness and an adversary could hence easily exploit his knowledge about the algorithm to compromise the confidentiality policy with the help of meta-inferences. This observation that additional distortions of per se harmless knowledge may be necessary to effectively enforce confidentiality policies is also well-known from other approaches to Controlled Interaction Execution enforcing confidentiality policies without lies [13, 15, 8, 10].

$$r = \{\, (a, b, c),\ (a, c, c),\ (b, a, c) \,\}$$

(a) Original database instance r

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

(b) Weakened view $\mathit{weak}(r, \mathit{psec})$ on r enforcing the confidentiality policy $\mathit{psec} = \{\, R(a, b, c),\ R(a, c, c) \,\}$

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

(c) Weakened view $\mathit{weak}(r, \mathit{psec}')$ on r enforcing the confidentiality policy $\mathit{psec}' = \{\, R(a, b, c),\ R(a, b, d) \,\}$

Figure 2.2: Enforcing simple confidentiality policies by weakening instances

2.2.2 Exemplifying the Simple Weakening Approach

To exemplify the algorithm for simplified input instances, consider the original database instance r given in Figure 2.2(a) and the confidentiality policy

$$\mathit{psec} = \{\, \Psi_1 = R(a, b, c),\ \Psi_2 = R(a, c, c) \,\}$$

both of whose potential secrets Ψ_1 and Ψ_2 are satisfied by the original instance r. The algorithm hence weakens this sensitive (definite) knowledge by replacing it by the weaker (but true) disjunctive knowledge $R(a, b, c) \vee R(a, c, c)$ and correspondingly returns the weakened view $\mathit{weak}(r, \mathit{psec})$ depicted in Figure 2.2(b).

An adversary can now try to draw conclusions about the (original) input instance, which led to this weakened view $\mathit{weak}(r, \mathit{psec})$, based on the knowledge this weakened view reveals about this input instance together with his general knowledge about the algorithm used to create the weakened view and his knowledge about the confidentiality policy *psec* set up for him. He can thereby gain

- the *positive knowledge* that (b, a, c) must be a tuple of this input instance,
- the *disjunctive knowledge* that at least one of the tuples (a, b, c) and (a, c, c) must be contained in this input instance and
- the *negative knowledge* that *no* other tuples are in this input instance.

The adversary can thus conclude that only one of the following (complete) input instances could have led to the released (incomplete) weakened view $\mathit{weak}(r, \mathit{psec})$:

- $r = \{ (a,b,c),\ (a,c,c),\ (b,a,c) \}$ (the original instance itself);
- $r^{\Psi_1} = \{ (a,c,c),\ (b,a,c) \}$ (an alternative instance obeying Ψ_1);
- $r^{\Psi_2} = \{ (a,b,c),\ (b,a,c) \}$ (an alternative instance obeying Ψ_2).

From the adversary's point of view each of these instances is *not* distinguishable from the original instance r used to construct $\mathit{weak}(r, \mathit{psec})$ because of

$$\mathit{weak}(r, \mathit{psec}) = \mathit{weak}(r^{\Psi_1}, \mathit{psec}) = \mathit{weak}(r^{\Psi_2}, \mathit{psec})$$

and might hence be the "real" input instance. As there are the instances r^{Ψ_1} obeying the potential secret Ψ_1 and r^{Ψ_2} obeying the potential secret Ψ_2 among these possibly "real" input instances, the adversary is *not* able to conclude that a specific policy element of psec is satisfied by the instance actually used to create $\mathit{weak}(r, \mathit{psec})$ and is hence *not* able to compromise the confidentiality policy psec.

As another example, again consider the original database instance r given in Figure 2.2(a) and the (different) confidentiality policy

$$\mathit{psec}' = \{ \Psi_1 = R(a,b,c),\ \Psi_2 = R(a,b,d) \}$$

containing only one potential secret satisfied by the original instance r. Similar to the above given example the algorithm replaces the knowledge about the satisfaction of Ψ_1 and the non-satisfaction of Ψ_2 by the weaker (but true) disjunctive knowledge $R(a,b,c) \vee R(a,b,d)$ and returns the weakened view $\mathit{weak}(r, \mathit{psec}')$ depicted in Figure 2.2(c).

An adversary can now again employ the knowledge this released weakened view $\mathit{weak}(r, \mathit{psec}')$ reveals about the original input instance together with his general knowledge about the weakening algorithm and the confidentiality policy psec' and can then conclude that only one of the following (complete) input instances could have led to the (incomplete) weakened view $\mathit{weak}(r, \mathit{psec}')$:

- $r = \{ (a,b,c),\ (a,c,c),\ (b,a,c) \}$ (the original instance itself, obeying Ψ_2);
- $r^{\Psi_1} = \{ (a,b,d),\ (b,a,c) \}$ (an alternative instance obeying Ψ_1);
- $r^{\Psi_2} = \{ (a,b,c),\ (b,a,c) \}$ (an alternative instance obeying Ψ_2).

Again, each of these instances is *not* distinguishable from the original instance r used to construct $weak(r, psec')$ from the adversary's point of view because of

$$weak(r, psec') = weak(r^{\Psi_1}, psec') = weak(r^{\Psi_2}, psec')$$

and might hence be the "real" input instance. As there are the instances r and r^{Ψ_2} both obeying the potential secret Ψ_2 as well as the instance r^{Ψ_1} obeying the potential secret Ψ_1 among these possibly "real" input instances, the adversary is again *not* able to conclude that a specific policy element of $psec'$ is satisfied by the instance actually used to create $weak(r, psec')$.

CHAPTER 3

A Generic Weakening Approach

So far, the basic ideas of achieving inference-proofness by weakening a database instance have been introduced. But for now, only simple confidentiality policies containing exactly two policy elements in the form of ground atoms have been considered. These basic ideas are now extended to be able to also deal with *non-simple* confidentiality policies containing an arbitrary number of policy elements in the more expressive form of existentially quantified atoms. This leads to the construction of a generic algorithm, which is proved to be confidentiality preserving and which is furthermore analyzed with respect to its complexity.

3.1 Clustering Non-Simple Confidentiality Policies

Until now, a simple confidentiality policy has essentially been enforced by (if necessary) constructing a disjunction containing both of its policy elements in the form of ground atoms. But in the more general case of non-simple confidentiality policies containing an arbitrary number of existentially quantified atoms, it is *not* desirable – neither in terms of availability, nor in terms of confidentiality – to construct disjunctions by arbitrarily grouping policy elements together.

Hence, dealing with non-simple confidentiality policies raises the question of how to construct a set of disjunctions, which is suitable for enforcing the confidentiality requirements specified by the non-simple policy without neglecting (implicit or explicit) availability requirements. This construction of disjunctions is to be discussed both on the *technical* level – dealing with how a given confidentiality policy can be enforced (independent of a specific application scenario) by a suitable set of disjunctions, thereby keeping availability as high as possible – as well as on the *ontological* level – dealing with structural properties each disjunction should have to be both meaningful in terms of availability and credible in terms of confidentiality with respect to a considered application scenario.

3.1.1 Basic Ideas of Clustering Potential Secrets

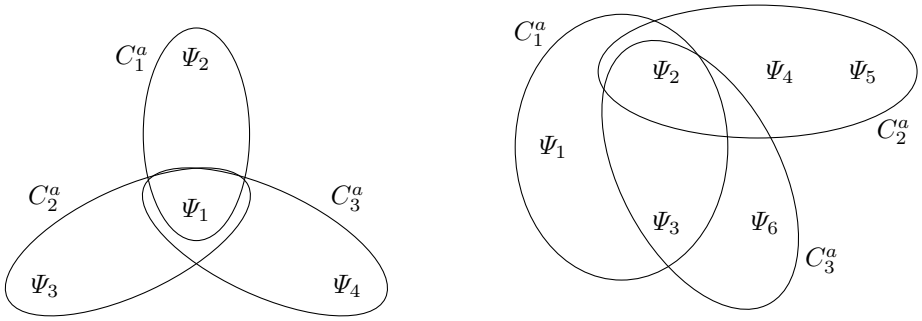
For now still restricting to policy elements in the form of ground atoms, a first *generic approach* to construct a suitable set of disjunctions enforcing a given confidentiality policy *psec* of arbitrary size relies on a partitioning of this policy *psec* into disjoint subsets called *clusters*. Each of these clusters C then induces a corresponding disjunction template $\bigvee_{\Psi \in C} \Psi$ and – in accordance with the basic ideas introduced in Chapter 2 – a weakened view on an original database instance r actually contains this disjunction $\bigvee_{\Psi \in C} \Psi$, if at least one of its disjuncts is satisfied by this instance r, i.e., $\mathcal{I}_r \models_M \Psi$.

But usually an arbitrary partitioning of a given confidentiality policy *psec* into a set of clusters is *not* desirable. Instead, a clustering of *psec* should be tailored to the needs of a specific application considered, thereby prudently balancing confidentiality and availability requirements induced by the characteristics of this application, as later discussed in more detail in Section 3.1.3. For the purpose of creating such an application-specific clustering, a so-called *notion of admissible indistinguishabilities* inducing a set $\mathcal{C}^a$ of *admissible clusters* should be provided. Two examples of such sets of admissible clusters are given in Figure 3.1.

Definition 3.1: Admissible Clusters

Given a confidentiality policy *psec*, the set $\mathcal{C}^a = \{C_1^a, \ldots, C_p^a\}$ with $C_i^a \subseteq psec$ and $|C_i^a| \geq 2$ for each $C_i^a \in \mathcal{C}^a$ is a set of *admissible clusters* over *psec*.

Given a set $\mathcal{C}^a$ of admissible clusters, each non-empty and non-singleton subset $C \subseteq C_i^a$ of potential secrets of an admissible cluster $C_i^a \in \mathcal{C}^a$ can be employed to construct the corresponding admissible disjunction template $\bigvee_{\Psi \in C} \Psi$. But when



(a) Set of induced admissible clusters of size 2 over the confidentiality policy $psec = \{\Psi_1, \Psi_2, \Psi_3, \Psi_4\}$

(b) Set of induced admissible clusters of size 3 over the confidentiality policy $psec = \{\Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Psi_6\}$

Figure 3.1: Examples of induced sets of admissible clusters

constructing a set of multiple disjunction templates covering each potential secret contained in an admissible cluster, care must be taken to ensure that these templates are pairwise disjoint as required above. This requirement becomes even more difficult when considering that admissible clusters do *not* need to be pairwise disjoint to make the task of defining such a set of admissible clusters – and hence also a security officer's duty to find a suitable notion of admissible indistinguishabilities – as easy as possible. Instead, the construction of a *disjoint* clustering $\mathcal{C}$, each of whose clusters is admissible, is the task of a clustering algorithm.

Definition 3.2: Clustering of a Confidentiality Policy

Suppose that $\mathcal{C}^a = \{C_1^a, \ldots, C_m^a\}$ is a set of admissible clusters induced by a given notion of admissible indistinguishabilities for a given confidentiality policy *psec*.

The set $\mathcal{C} = \{C_1, \ldots, C_q\}$ is a *clustering* of *psec* obeying $\mathcal{C}^a$, if

(i) for each cluster $C_j \in \mathcal{C}$ there is

- an admissible cluster $C_i^a \in \mathcal{C}^a$ with $C_j \subseteq C_i^a$ or
- C_j is a singleton cluster $C_j = \{\Psi\}$ with $\Psi \in psec$,

(ii) $C_i \cap C_j = \emptyset$ for all pairs of different clusters $C_i, C_j \in \mathcal{C}$ and

(iii) for each potential secret $\Psi \in psec$ there is a cluster $C_j \in \mathcal{C}$ with $\Psi \in C_j$.

This definition obviously guarantees that each potential secret is contained in exactly one cluster and that a clustering is hence pairwise disjoint. To also allow the handling of policy elements, for which the considered notion of admissible indistinguishabilities (in combination with the given confidentiality policy) does *not* induce an admissible cluster (of non-trivial size[1]), each singleton subset of the considered confidentiality policy is always supposed to be a valid (but trivial) cluster. But even under the supposition that each policy element is contained in a (non-trivial) admissible cluster, it might nonetheless *not* be possible to construct a disjoint clustering of only non-singleton clusters.

To exemplify the latter case, consider the set of admissible clusters of size 2, which is given in Figure 3.1 (a). Each of the potential secrets Ψ_2, Ψ_3 and Ψ_4 is contained in only one admissible cluster such that each of these policy elements can only admissibly be paired with the potential secret Ψ_1. Hence, a clustering of only non-singleton clusters covering each element of the given confidentiality policy $psec = \{ \Psi_1, \Psi_2, \Psi_3, \Psi_4 \}$ can *not* be constructed without violating the requirement that a clustering should be disjoint.

In case of the set of admissible clusters of size 3 given in Figure 3.1 (b), a valid clustering of only non-singleton clusters can instead easily be constructed. An example of such a clustering is $\mathcal{C} = \{C_1, C_2, C_3\}$ with

$$C_1 = \{ \Psi_1, \Psi_2 \} \subseteq C_1^a, \;\; C_2 = \{ \Psi_4, \Psi_5 \} \subseteq C_2^a \quad \text{and} \quad C_3 = \{ \Psi_3, \Psi_6 \} \subseteq C_3^a \;\; .$$

This clustering of only non-singleton clusters is valid, as it consists of pairwise disjoint clusters in the form of subsets of admissible clusters together covering each potential secret of the given policy $psec = \{ \Psi_1, \ldots, \Psi_6 \}$. But under the supposition that clusters of a size larger than 2 are required (as discussed below), one can easily see that neither of the two example sets of admissible clusters given in Figure 3.1 allow for the construction of such a clustering.

Generally, each singleton cluster $C_j = \{\Psi\}$ of a clustering $\mathcal{C}$ is *not* suitable for enforcing the potential secret Ψ, as it induces the trivial disjunction template $\bigvee_{\bar{\Psi} \in C_j} \bar{\Psi} = \Psi$ revealing definite knowledge about the satisfaction of Ψ. In fact, each disjunction must consist of at least two (semantically pairwise different) disjuncts to be able to hide which of its disjuncts are actually satisfied by a considered original database instance. Then, an adversary only knows that a non-empty subset of these disjuncts is satisfied by this instance.

In some application scenarios it might be even worthwhile to require disjunctions of a length longer than 2 – and hence also clusters of a corresponding size – to

[1] Note that a singleton set of policy elements possibly induced by a notion of admissible indistinguishabilities is *not* considered to be an admissible cluster according to Definition 3.1.

correspondingly increase this number of non-empty subsets of policy elements, which might – from an adversary's point of view – possibly be satisfied by a considered original instance. This decreases the knowledge an adversary can gain about this instance and hence increases the achieved level of confidentiality.

As exemplified above, it might sometimes *not* be possible to construct a clustering consisting only of clusters of a certain minimum size of k^* with $k^* \geq 2$. To nonetheless guarantee a wanted level of confidentiality, so-called *additional potential secrets* are (artificially) introduced to allow for the construction of clusters *not* smaller than k^*. Such a clustering, which is (possibly) extended by additional potential secrets, is referred to as an extended clustering $\mathcal{C}^*$. Thereby, an additional potential secret is an existentially quantified atom (or a ground atom, as assumed in this subsection) just like any non-additional potential secret stemming from the given confidentiality policy.

Definition 3.3: Extended Clustering of a Confidentiality Policy

Let *psec* be a confidentiality policy and suppose that a notion of admissible indistinguishabilities is given. Further, suppose that $k^* \in \mathbb{N}$ with $k^* \geq 2$ is the minimum size each (extended) cluster should have.

The set $\mathcal{C}^* = \{C_1, \ldots, C_q\}$ is an *extended clustering* of *psec* obeying the given notion of admissible indistinguishabilities and k^*, if

(i) each (extended) cluster $C_j \in \mathcal{C}^*$ is a set of existentially quantified atoms and has a minimum size of k^*, i.e., $|C_j| \geq k^*$,

(ii) each (extended) cluster $C_j \in \mathcal{C}^*$ contains at least one original policy element, i.e., $C_j \cap \mathit{psec} \neq \emptyset$ for each $C_j \in \mathcal{C}^*$,

(iii) for each (extended) cluster $C_j \in \mathcal{C}^*$ each existentially quantified atom $\Psi^A \in C_j$ with $\Psi^A \notin \mathit{psec}$ is an additional potential secret,

(iv) the given notion of admissible indistinguishabilities induces a set $\mathcal{C}^a$ of admissible clusters for the set $\bigcup_{C_j \in \mathcal{C}^*} C_j$ of all (possibly additional) potential secrets such that $\mathcal{C}^*$ is a clustering of $\bigcup_{C_j \in \mathcal{C}^*} C_j$ obeying $\mathcal{C}^a$ according to Definition 3.2 and

(v) the set $\mathit{psec}^A := \bigcup_{C_j \in \mathcal{C}^*} C_j \setminus \mathit{psec}$ is an admissible set of additional potential secrets (still to be defined).

Requirement (iv) of this definition guarantees that each constructed cluster – including those containing additional potential secrets – is admissible with respect

to a considered notion of admissible indistinguishabilities. As a direct consequence, such a notion might – dependent on the inputs of the clustering algorithm and hence also on the employed notion of admissible indistinguishabilities itself – need to be able to also handle a certain set of additional potential secrets to enable the clustering algorithm to construct a valid extended clustering. A detailed discussion of the requirements such a *well-defined* notion of admissible indistinguishabilities should meet is given in Section 3.1.3. Additionally, condition (v) of Definition 3.3 deliberately leaves open which further requirements an *admissible* set of additional potential secrets has to fulfill to ensure that the given definition of an extended clustering is "generic enough" to be also applicable within the extended scenario handling potential secrets in the form of existentially quantified atoms (instead of only ground atoms), which is later discussed in Section 3.1.2.

If an employed notion of admissible indistinguishabilities allows for the construction of an extended clustering $\mathcal{C}^*$ such that each extended cluster $C \in \mathcal{C}^*$ contains only one non-additional potential secret and $|C| - 1$ additional potential secrets, such an extended clustering $\mathcal{C}^*$ is a trivial but feasible solution from the point of view of a clustering algorithm. But such a trivial solution is usually *not* desirable in terms of availability, as each constructed additional potential secret specifies some additional knowledge, whose validity must *not* be revealed to an adversary. Hence, usually the goal is to find an extended clustering $\mathcal{C}^*$ containing only a minimum number of additional potential secrets and to thereby maximize availability within the limits given by the confidentiality requirements.

In general, the construction of an extended clustering should *not* depend on a given original database instance r, although such an instance-dependent construction might open up extended possibilities to construct availability-maximizing clusterings. For example, an instance-dependent clustering algorithm could deliberately maximize the number of (possibly additional) potential secrets grouped to clusters, whose corresponding disjunction templates are *not* satisfied by a considered original instance. Thereby, this algorithm could increase the number of potential secrets, whose real truth values do *not* need to be kept confidential and are hence revealed to an adversary. On the downside, such an instance-dependent clustering must take additional care to ensure that each alternative instance, which should be indistinguishable from the original instance from an adversary's point of view, leads to exactly the same clustering – and hence also to the same weakened view – as the original instance to mitigate harmful meta-inferences an adversary might use to exclude some alternative instances from being credible.

To eliminate another source for meta-inferences, an artificially introduced additional potential secret should generally be treated just like any non-additional potential secret when constructing a weakened view. This disables an adversary

of taking advantage of his knowledge about which of the disjuncts of a published weakened view are artificially introduced. This knowledge is supposed to be publicly available as trying to keep this knowledge confidential is *not* a promising approach anyway: an adversary is supposed to be aware of the clustering algorithm as well as of the confidentiality policy and the employed notion of admissible indistinguishabilities and is hence able to determine the computed extended clustering – including the construction of additional potential secrets – himself by simulating the computations of the protection mechanism.

3.1.2 Dealing with Existentially Quantified Potential Secrets

Until now, the enforcement of confidentiality policies with the help of disjunctions has been discussed under the assumption that a confidentiality policy only contains potential secrets in the form of ground atoms. But in fact, a confidentiality policy may also contain more expressive sentences in the form of existentially quantified atoms according to Definition 2.4. While this clearly improves the expressiveness of confidentiality policies on the one hand, it might also open up new possibilities for an adversary to draw harmful inferences on the other hand.

To realize that there is indeed such a new possibility to draw harmful inferences, consider the following confidentiality policy

$$psec = \{\, R(a,b,c),\ R(a,b,d),\ (\exists X)\, R(a,b,X),\ (\exists X)\, R(b,b,X) \,\}$$

containing existentially quantified potential secrets. Assuming that disjunction templates of a minimum length of 2 are required, the clustering sketched so far might lead to a weakened view containing the disjunction $R(a,b,c) \vee R(a,b,d)$. But this disjunction directly implies the knowledge $(\exists X)\, R(a,b,X)$, which itself is protected by a potential secret of the policy *psec*. Hence, an adversary – to whom this disjunction is revealed as true knowledge about the original instance – is able to violate the confidentiality policy *psec*, as there is *no* alternative database instance, which satisfies the considered disjunction $R(a,b,c) \vee R(a,b,d)$ and which furthermore obeys the potential secret $(\exists X)\, R(a,b,X)$.

More generally, this example indicates that the clustering sketched so far might induce disjunctions implying knowledge, which itself is protected by potential secrets of the confidentiality policy. So, this implied – and hence weaker – knowledge is still too strong. A possibility to avoid the construction of these too strong disjunctions is to first *clean* the given confidentiality policy in a preprocessing step and then subsequently apply the clustering discussed above. Thereby, cleaning a confidentiality policy means to reduce this policy to a maximum "core" subset

of its weakest sentences and hence results in a set of potential secrets which do pairwise *not* imply each other.[2]

Definition 3.4: Cleaned Set

Let $\mathcal{S}$ be a set of sentences of the first-order language $\mathscr{L}$. Its *cleaned set* $\widehat{\mathcal{S}}$ is a maximum subset of weakest sentences of $\mathcal{S}$ such that no pair of different sentences of $\widehat{\mathcal{S}}$ is semantically equivalent.

A sentence $\Psi \in \mathcal{S}$ is a *weakest sentence* of $\mathcal{S}$, if for each sentence $\Psi' \in \mathcal{S}$

(i) either the implication $\Psi' \models_{DB} \Psi$ holds or

(ii) both non-implications $\Psi' \not\models_{DB} \Psi$ and $\Psi \not\models_{DB} \Psi'$ hold.

On the operational level a cleaned set $\widehat{\mathcal{S}}$ of a set $\mathcal{S}$ of sentences can be computed by initially setting $\widehat{\mathcal{S}} := \mathcal{S}$ and then repeatedly removing an arbitrary sentence Ψ' from $\widehat{\mathcal{S}}$, for which there is a sentence $\Psi \in \widehat{\mathcal{S}}$ with $\Psi' \neq \Psi$ and $\Psi' \models_{DB} \Psi$. This sentence Ψ' can be safely removed from $\widehat{\mathcal{S}}$ as it is stronger than Ψ or semantically equivalent to Ψ and must hence *not* be in the cleaned set. The algorithm finally terminates as soon as there is *no* such sentence $\Psi' \in \widehat{\mathcal{S}}$ anymore. Then, for each pair of (syntactically) different sentences $\Psi, \Psi' \in \widehat{\mathcal{S}}$ there is *no* semantic equivalence between Ψ and Ψ' and there is also *no* implication relationship between Ψ and Ψ' as required by condition (ii) of Definition 3.4 – and hence each remaining sentence of $\widehat{\mathcal{S}}$ is also a weakest sentence of the non-cleaned set $\mathcal{S}$. As a consequence, $\widehat{\mathcal{S}}$ is a cleaned set, as it is moreover maximum because adding an arbitrary sentence of $\mathcal{S} \setminus \widehat{\mathcal{S}}$ would result in a violation of the properties of a cleaned set.

Reconsidering the example policy *psec* given above, the corresponding cleaned confidentiality policy is

$$\widehat{psec} = \{\, (\exists X)\, R(a, b, X),\ (\exists X)\, R(b, b, X) \,\} \ .$$

Obviously, both policy elements of this cleaned policy $\widehat{psec}$ do *not* imply each other. Moreover, the potential secrets $R(a, b, c)$ and $R(a, b, d)$ of the original policy *psec* are *not* taken over into the cleaned policy $\widehat{psec}$, as the original policy *psec* also contains the (weaker) sentence $(\exists X)\, R(a, b, X)$ implied by both of the (stronger) sentences $R(a, b, c)$ and $R(a, b, d)$.

[2] Note that weakest sentences can equivalently be found by considering "$\models_{DB}$" as a partial preorder on existentially quantified atoms – in the sense that $\Psi \leq \Psi'$ is supposed to hold, *if and only if* $\Psi' \models_{DB} \Psi$ holds – and then taking the minimal elements (cf. [52]).

Assuming that the disjunction $(\exists X)\, R(a,b,X) \vee (\exists X)\, R(b,b,X)$ is constructed to enforce the cleaned policy $\widehat{psec}$, this disjunction does *not* imply any (weaker) knowledge, which itself is protected by the cleaned policy $\widehat{psec}$ or the original policy *psec*. In particular, even the potential secrets $R(a,b,c)$ and $R(a,b,d)$, which are only contained in the original policy *psec* and *not* in cleaned policy $\widehat{psec}$, are protected by this disjunction: from an adversary's point of view the existence of an alternative instance r' with

$$\mathcal{I}_{r'} \models_M (\exists X)\, R(a,b,X) \vee (\exists X)\, R(b,b,X) \quad \text{and} \quad \mathcal{I}_{r'} \not\models_M (\exists X)\, R(a,b,X)$$

is possible and for such an instance r' also $\mathcal{I}_{r'} \not\models_M R(a,b,c)$ and $\mathcal{I}_{r'} \not\models_M R(a,b,d)$ has to hold – otherwise $\mathcal{I}_{r'} \models_M (\exists X)\, R(a,b,X)$ would hold, too.

This implicit protection of all of those policy elements of the original policy *psec*, which are *not* taken over into the cleaned policy $\widehat{psec}$, by disjunctions enforcing the potential secrets of $\widehat{psec}$ can be generalized as follows.

Lemma 3.1: Implicit Protection

Let Ψ_S and Ψ_W be sentences of $\mathscr{L}$ such that Ψ_W is at least as weak as 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., $\mathcal{I}_r \not\models_M \Psi_S$.

Proof. According to the definition of DB-Implication, $\Psi_S \models_{DB} \Psi_W$ *if and only if* for each DB-Interpretation $\mathcal{I}$ with $\mathcal{I} \models_M \Psi_S$ also $\mathcal{I} \models_M \Psi_W$. By contraposition, this is equivalent to $\Psi_S \models_{DB} \Psi_W$ *if and only if* for each DB-Interpretation $\mathcal{I}$ with $\mathcal{I} \not\models_M \Psi_W$ also $\mathcal{I} \not\models_M \Psi_S$. So, $\mathcal{I}_r \not\models_M \Psi_W$ directly implies $\mathcal{I}_r \not\models_M \Psi_S$ under the assumption that $\Psi_S \models_{DB} \Psi_W$ is given. ♠

As known from Section 3.1.1, it is *not* always possible to construct a disjoint clustering, which solely consists of potential secrets stemming from the original confidentiality policy. Instead, it may be necessary to extend too small clusters by creating so-called additional potential secrets. As a direct consequence, care must be taken to ensure that the disjunctions resulting from these extended clusters do *not* provide too strong knowledge compromising the confidentiality policy. Hence, adopting the counter measure discussed above, additional potential secrets should be constructed such that the union of all additional potential secrets and all non-additional potential secrets satisfies the properties of a cleaned set. This leads to the following criterion for an admissible set of additional potential secrets, which implements the purely generic requirement an extended clustering has to satisfy according to condition (v) of Definition 3.3.

Definition 3.5: Admissible Additional Potential Secrets

Let $\widehat{psec}$ be a cleaned confidentiality policy. A set $\widehat{psec}^A$ of additional potential secrets is *admissible* with respect to the given confidentiality policy $\widehat{psec}$, if the union $\widehat{psec} \cup \widehat{psec}^A$ is a cleaned set.

As a direct consequence of this definition relying on the definition of a cleaned set, there is *no* implication relationship – and hence also *no* semantic equivalence – between each pair of different (possibly additional) potential secrets of an extended clustering. As a further consequence, the confidentiality policy *psec* must *not* contain a potential secret Ψ_W, which is semantically equivalent to the weakest possible potential secret $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$ *not* containing any constant symbols. Otherwise, the corresponding cleaned confidentiality policy $\widehat{psec}$ would solely consist of this weakest policy element Ψ_W – as all other potential secrets of *psec* would imply this weakest policy element Ψ_W and would hence *not* be taken over into the cleaned policy – and no admissible additional potential secret Ψ_W^A *not* implying Ψ_W could be constructed for Ψ_W.

According to Definition 3.3, each cluster of an extended clustering must obey a given notion of admissible indistinguishabilities. This, in particular, also applies for clusters containing additional potential secrets. Hence, the construction of an admissible set of additional potential secrets is closely related to a reasonable notion of admissible indistinguishabilities, as discussed in the following.

3.1.3 About Admissible Indistinguishabilities

As already mentioned in Section 3.1.1, the quality of a weakening crucially relies on the employed notion of admissible indistinguishabilities, which induces the set of admissible clusters and thereby restricts the construction of admissible disjunction templates. Although the purpose of disjunctions is to protect information by weakening an adversary's knowledge about the original instance, each disjunction should still provide as much useful information as possible in terms of availability – otherwise the enforcement of confidentiality policies by disjunctions might *not* provide any advantage over the (traditional) complete refusal of protected information. In terms of confidentiality all alternatives provided by a disjunction should moreover be equally probable from an adversary's point of view. Otherwise, this adversary might be able to strengthen his knowledge about the original instance by excluding those alternatives provided by a disjunction, whose validity seems to be highly unlikely from a practical point of view. Hence, it is of crucial

importance that the employed notion of admissible indistinguishabilities fits the specific application scenario considered.

Obviously, there is *no* general notion of admissible indistinguishabilities that fits each possible application scenario and there is also *no* generally valid approach to find such a notion for a specific application scenario. Instead, it is the duty of a security officer to analyze the characteristics of such a specific application scenario with scrutiny and to subsequently develop a suitable notion of admissible indistinguishabilities, which prudently balances confidentiality and availability. 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 clusters resulting from such a notion of admissible indistinguishabilities, a generic method to construct admissible clusters based on a high level specification language is needed.

Usually, a confidentiality policy should be managed by a database system implementing well-known and widespread query languages such as SQL or relational algebra [75, 78, 1]. As an admissible cluster induced by a notion of admissible indistinguishabilities is essentially a subset of potential secrets, a security officer can create a set of admissible clusters of size k by computing a series of $k-1$ self-joins on the database table managing the confidentiality policy. Then, the security officer's concrete notion of admissible indistinguishabilities corresponds to the implemented join conditions. But of course, a security officer is free to use any other programming language, which is expressive enough.

As already known, there might be the need to introduce additional potential secrets to be able to construct an extended clustering. But the definition of an extended clustering requires that the potential secrets of each of its clusters – without making a difference between non-additional potential secrets stemming from the confidentiality policy and additional potential secrets *not* stemming from the confidentiality policy – form a valid cluster according to the given notion of admissible indistinguishabilities. Hence, considering a specific confidentiality policy and a minimum size k^* each (extended) cluster should have, it may *not* be sufficient that such a given notion of admissible indistinguishabilities is able to handle only the non-additional potential secrets of the considered confidentiality policy. Instead, this notion must be "generic enough" to also handle a certain set of additional potential secrets allowing for the construction of at least one extended clustering for the considered confidentiality policy.

In a worst-case scenario up to k^*-1 additional potential secrets are needed for each non-additional potential secret of the cleaned confidentiality policy to construct an extended clustering with clusters of size k^*. To later free the proof of inference-proofness from the burden of discussing non-determinism, each additional potential secret should be constructed by a deterministic algorithm. This construction

of such an additional potential secret might require the use of additional constant symbols *not* occurring in any original policy element, but should only use constant symbols of the domain Dom of $\mathscr{L}$ to guarantee that each constructed additional potential secret – in the form of an existentially quantified atom – is a sentence of the employed first-order language $\mathscr{L}$. Hence, additionally considering that an extended clustering $\mathcal{C}^*$ contains a finite number of (possibly additional) potential secrets each of which is of finite arity, the active domain of $\mathcal{C}^*$ – consisting of the union of all constant symbols occurring in a (possibly additional) potential secret of $\mathcal{C}^*$ – should be a finite subset of the domain Dom.

Definition 3.6: Well-Defined Indistinguishability

Let $\widehat{psec}$ be the cleaned set of a given confidentiality policy $psec$. Further, suppose that $k^* \in \mathbb{N}$ with $k^* \geq 2$ is the minimum size each (extended) cluster of an extended clustering should have.

A notion of admissible indistinguishabilities is *well-defined* with respect to $psec$ and k^*, if there is an extended clustering $\mathcal{C}^*$ of $\widehat{psec}$ obeying the considered notion of admissible indistinguishabilities and k^* according to Definition 3.3 such that

(i) there is a deterministic algorithm creating this extended clustering $\mathcal{C}^*$ with all of its additional potential secrets Ψ^A with $\Psi^A \notin \widehat{psec}$,

(ii) the active domain of $\mathcal{C}^*$ is a finite subset of the domain Dom and

(iii) the domain Dom contains at least one constant symbol, which is *not* contained in the active domain of the extended clustering $\mathcal{C}^*$.

Reconsidering that the active domain $ad(\mathcal{C}^*)$ of $\mathcal{C}^*$ is a *finite* set and that the domain Dom of $\mathscr{L}$ is supposed to be *infinite*, requirement (iii) of this definition seems to be always satisfied at first sight because of $|Dom| > |ad(\mathcal{C}^*)|$. But while this assumption of an *infinite* domain – together with the assumption of a database system providing only untyped attributes (cf. [18, 75, 78]) – is adequate in theory, it is usually *not* realizable in practice: technical limitations such as finite memory or the finite number of bits a data type provides to represent values limit the overall number of different representable values. Moreover, the considered application scenario might also limit the number of reasonable values for a specific column of a database instance. For example, a column representing the biological sex of a person should usually only contain the values `male` and `female`, even if the data type of the corresponding column offers a much larger domain.

As already discussed in Section 3.1.2 and later elaborated in more detail in Section 3.3, the inference-proofness of a weakened view crucially relies on clustering only (possibly additional) potential secrets *not* implying each other to disjunction templates. Thereby, this notion of non-implication (employed in the definition of a cleaned set) relies on DB-Implication, which in turn relies on DB-Interpretations employing the *infinite* domain *Dom* as their universe. But if instead only a "small enough" *finite* domain is considered for a DB-Interpretation, an adversary might be able to draw confidentiality compromising conclusions regarding the satisfaction of a potential secret based on the satisfaction of other potential secrets with the help of so-called combinatorial effects.

To exemplify that an adversary might indeed take advantage of such a "small enough" *finite* domain, suppose that the (extended) clusters

- $C_1 = \{\, (\exists X)\, R(a, X, a),\ (\exists X)\, R(a, X, b)\,\}$ and
- $C_2 = \{\, (\exists X)\, R(a, a, X),\ (\exists X)\, R(a, b, X)\,\}$

constitute an extended clustering $\mathcal{C}^*$ of a given confidentiality policy. This policy is cleaned and does hence *not* contain any implication relationships between pairs of different potential secrets in terms of usual DB-Implication. Additionally considering an original database instance r, which satisfies a potential secret of the cluster C_1 but *not* any potential secret of the cluster C_2, the released weakened view contains the disjunction

$$(\exists X)\, R(a, X, a) \vee (\exists X)\, R(a, X, b)\ \ ,$$

but does *not* contain the disjunction template

$$(\exists X)\, R(a, a, X) \vee (\exists X)\, R(a, b, X)\ \ .$$

An adversary is hence supposed to know that the original instance does *neither* satisfy $(\exists X)\, R(a, a, X)$ *nor* $(\exists X)\, R(a, b, X)$.

Further assuming that $Dom = \{a, b\}$ is the underlying *finite* domain, the disjunct $(\exists X)\, R(a, X, b)$ of the constructed disjunction (corresponding to C_1) can only be satisfied by an alternative instance containing (at least) one of the tuples (a, a, b) or (a, b, b). But an adversary can exclude such an alternative instance from being the original instance r, as such an instance would – in contrast to the original instance r – satisfy at least one of the potential secrets of the cluster C_2. Hence, the disjunct $(\exists X)\, R(a, X, b)$ can *not* be satisfied by any (alternative) instance and the adversary can consequently employ the constructed disjunction to conclude that the potential secret $(\exists X)\, R(a, X, a)$ must be satisfied by the original instance r, thereby compromising the confidentiality policy.

If the considered *finite* domain *Dom* is extended by an additional "fresh" constant symbol c *not* occurring in the active domain of the clustering $\mathcal{C}^*$ – thereby satisfying requirement (iii) of Definition 3.6 – the alternative instances

- $r' = \{\, (a, c, a) \,\}$ obeying the potential secret $(\exists X)\, R(a, X, b)$ and
- $r'' = \{\, (a, c, b) \,\}$ obeying the potential secret $(\exists X)\, R(a, X, a)$,

which both satisfy the constructed disjunction and both do *not* satisfy any potential secret of the cluster C_2, can easily be constructed. Hence, an adversary is *not* able to compromise the confidentiality policy under this extended *finite* domain, as the additional "fresh" constant symbol breaks up the above mentioned implication relationships between policy elements, which are based on combinatorial effects possible because of a too small *finite* domain.

So, although favoring an *infinite* domain in theory to avoid combinatorial effects possibly leading to harmful inferences, a "sufficiently large" *finite* domain is adequate in practice. When later showing the inference-proofness of the approach in Section 3.3, a so-called ground operator is employed. This ground operator is shown to be well-defined in Lemma 3.2 and the supposition that only one "fresh" constant symbol is sufficient to disable any implication relationships within a cleaned set is a direct consequence of the proof of this lemma. But in terms of the credibility of these non-implications from an adversary's point of view – and hence also in terms of the credibility of alternative instances – a much larger supply of "fresh" constant symbols, which moreover fit the (different) domains of the attributes occurring in the database schema as well as the semantic context of the application scenario, is of course highly desirable.

3.2 Construction of Weakened Views

Now that the challenge of clustering a confidentiality policy to a set of pairwise disjoint admissible disjunction templates is succeeded, these disjunction templates can be employed to enforce this confidentiality policy by constructing a suitably weakened view on a given (original) database instance.

Similar to the simple weakening algorithm, which is sketched in Section 2.2.1 and aims at the construction of a weakened view enforcing a simplified confidentiality policy, the basic idea is to replace each confidentiality compromising tuple of a given original database instance by the subset of all those disjunction templates, which are implied by this database tuple. Thereby, such a database tuple is considered to be confidentiality compromising, if it implies at least one disjunction template. Otherwise, this tuple is considered to be harmless and the validity of

this tuple is revealed. As a consequence, the knowledge about the validity of those policy elements, which are actually satisfied by the original database instance, is weakened by corresponding disjunctions and an adversary is hence *not* able to infer which of the alternatives provided by a disjunction is actually satisfied by the original database instance.

The construction of the simple weakening algorithm in Section 2.2.1 suggests that a weakened view should also reveal the knowledge about the non-validity of those constant combinations over the considered database schema, which are actually *not* contained in the original database instance and whose exact validity status is *not* deliberately left unknown to enforce the confidentiality policy. For this purpose, the simple weakening algorithm introduces a so-called (partial) completeness sentence. This concept can be adapted for potential secrets (and hence disjuncts) with existentially quantified variables by constructing a (partial) completeness sentence exposing the non-validity of each constant combination, which *neither* corresponds to a harmless positive ground atom of the weakened view *nor* induces a DB-Interpretation satisfying a disjunction of the weakened view.

But if a confidentiality policy does *not only* consist of ground atoms, such a completeness sentence might *not* capture all negative knowledge an adversary is actually aware of. For example, consider an extended clustering inducing the disjunction templates

- $(\exists X)\, R(b, X, e) \vee (\exists X)\, R(b, X, d)$ and
- $(\exists X)\, R(b, e, X) \vee (\exists X)\, R(c, e, X)$

and further suppose that an original database instance r is given, which satisfies the first of these disjunction templates, but *not* the second one. Then, in accordance with the basic algorithmic ideas of Section 2.2.1, the weakened view on r contains only the first of the above given disjunctions and the completeness sentence of this weakened view correspondingly excludes each constant combination satisfying one of the disjuncts $(\exists X)\, R(b, X, e)$ and $(\exists X)\, R(b, X, d)$ of the satisfied disjunction template from being non-valid.

As an adversary is supposed to be aware of both the confidentiality policy as well as of the algorithm used to construct the weakened view and as the construction of an extended clustering does further *not* depend on the considered original database instance, this adversary can simulate the construction of the extended clustering, whose corresponding disjunction templates are given above. Additionally considering that the released weakened view does *not* contain the second disjunction template $(\exists X)\, R(b, e, X) \vee (\exists X)\, R(c, e, X)$, the adversary can again employ his knowledge about the weakening algorithm to reason that neither $(\exists X)\, R(b, e, X)$ nor $(\exists X)\, R(c, e, X)$ is satisfied by the original database instance – as otherwise

the corresponding disjunction template would have been added to the weakened view as a disjunction.

So, although the adversary knows that each constant combination $(b, e, \square)$ with $\square$ being an arbitrary constant symbol of *Dom* is *not* valid according to the original database instance, this knowledge is *not* reflected properly by the (partial) completeness sentence of the constructed weakened view. In particular, the adversary knows that the constant combination (b, e, e) is *not* satisfied, but this knowledge is *not* reflected by the completeness sentence as this sentence explicitly excludes each constant combination satisfying $(\exists X)\, R(b, X, e)$ from being non-valid.

More generally, this problem of a (partial) completeness sentence *not* reflecting an adversary's negative knowledge properly arises, if

- an extended clustering contains one cluster, whose corresponding disjunction $\Psi_1 \vee \ldots \vee \Psi_k$ is satisfied by a considered original database instance, and another cluster, whose corresponding disjunction $\bar{\Psi}_1 \vee \ldots \vee \bar{\Psi}_{\bar{k}}$ is *not* satisfied by this instance and
- these disjunctions contain disjuncts Ψ_i and $\bar{\Psi}_j$ such that there is a constant combination over the database schema of the original instance, which satisfies both Ψ_i and $\bar{\Psi}_j$.

To mitigate this problem – and to hence take care that a weakened view reflects all knowledge an adversary actually has – the (partial) completeness sentence of a weakened view should be complemented by each *negated disjunction* $\neg\,[\,\bigvee_{\Psi \in C} \Psi\,]$ corresponding to a cluster C of the constructed extended clustering, which is *not* satisfied by the original database instance.

These considerations extending the basic ideas of the simple weakening algorithm introduced in Section 2.2.1 finally lead to the following construction of weakened views. As also motivated in Section 2.2.1, such a weakened view should consist of ordered sequences of sentences to suitably normalize these weakened views and to thereby prevent an adversary from drawing harmful meta-inferences on the basis of the syntactic appearance of weakened views.

Definition 3.7: Weakened View

Let r be a complete database instance over a database schema $\langle R|\mathcal{A}_R|\,\emptyset\,\rangle$. Further, suppose that $\mathcal{C}_r^*$ is the subset of those clusters of an extended clustering $\mathcal{C}^*$ of a cleaned confidentiality policy $\widehat{psec}$ such that $\mathcal{I}_r \models_M \bigvee_{\Psi \in C} \Psi$ holds for each cluster $C \in \mathcal{C}_r^*$.

Then, the *weakened view* $weak(r, psec)$ on r consists of the following totally ordered sequences of sentences of the first-order language $\mathscr{L}$:

(i) *Positive knowledge* $weak(r, psec)^+$: Each tuple $\boldsymbol{c} \in r$, for which the non-implication $R(\boldsymbol{c}) \not\models_{DB} \Psi$ holds for each (possibly additional) potential secret $\Psi \in \bigcup_{C \in \mathcal{C}_r^*} C$, is modeled as the ground atom $R(\boldsymbol{c})$.

(ii) *Disjunctive knowledge* $weak(r, psec)^\vee$: For each cluster $C \in \mathcal{C}_r^*$ the disjunction $\bigvee_{\Psi \in C} \Psi$ is constructed.

(iii) *Negative knowledge* $weak(r, psec)^-$: For each cluster $C \in (\mathcal{C}^* \setminus \mathcal{C}_r^*)$ the negated disjunction $\neg [\bigvee_{\Psi \in C} \Psi]$ is constructed. Moreover, a (partial) completeness sentence having a universally quantified variable X_j for each attribute $A_j \in \mathcal{A}_R$ is built. This sentence is supposed to contain 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
- each disjunct $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ occurring in $weak(r, psec)^\vee$

and finally contains $\neg R(X_1, \ldots, X_n)$ as its last disjunct.

Under the supposition that there is a total order on the set of those (constant and variable) symbols, which might appear as terms of sentences of a weakened view, a total order on the sentences actually occurring within a weakened view $weak(r, psec)$ can for example be established as follows:

- the presentation of the weakened view starts with all ground atoms of the positive knowledge $weak(r, psec)^+$ and the sequence of these ground atoms is sorted lexicographically according to the order on their constant symbols;
- then, all disjunctions of the disjunctive knowledge $weak(r, psec)^\vee$ follow such that first within each of these disjunctions the sequence of its disjuncts is sorted lexicographically according to the order on their terms and subsequently the sequence of all of these disjunctions is sorted lexicographically according to the order on their terms;
- after that all negated disjunctions of the negative knowledge $weak(r, psec)^-$ are presented and the sequence of (and within) these sentences is ordered just as the disjunctions of $weak(r, psec)^\vee$;
- and finally, the (partial) completeness sentence of the negative knowledge $weak(r, psec)^-$ is given and normalized as follows: first, within each of its disjuncts of the form $(\bigwedge_{i \in \{1,\ldots,n\} \text{ with } t_i \in Dom} X_i \equiv t_i)$ the sequence of the conjuncts of the form $X_i \equiv t_i$ is sorted lexicographically according to the

order on the variable symbols X_i and after that the sequence of these disjuncts is sorted lexicographically according to the order on their terms; then, $\neg R(X_1, \ldots, X_n)$ is finally appended as the last disjunct.

A detailed example of such a construction of a weakened view is later given in Figure 4.4, when discussing a concrete (availability-maximizing) instantiation of the generic weakening algorithm in Chapter 4.

According to the above given construction of weakened views, the knowledge represented by a weakened view $\mathit{weak}(r, \mathit{psec})$ is completely true with respect to the considered original database instance r, i.e., $\mathcal{I}_r \models_M \mathit{weak}(r, \mathit{psec})$: for each ground atom $R(\boldsymbol{c})$ of the positive knowledge $\mathit{weak}(r, \mathit{psec})^+$ the tuple $\boldsymbol{c}$ is actually contained in the original instance r and hence $\mathcal{I}_r \models_M \mathit{weak}(r, \mathit{psec})^+$ is a direct consequence. Moreover, each disjunction $\bigvee_{\Psi \in C} \Psi$ of the disjunctive knowledge $\mathit{weak}(r, \mathit{psec})^\vee$, which is constructed for a cluster $C \in \mathcal{C}_r^*$, contains at least one disjunct satisfied by $\mathcal{I}_r$ – otherwise, the cluster C would *not* be in $\mathcal{C}_r^*$ according to its definition. This immediately results in $\mathcal{I}_r \models_M \mathit{weak}(r, \mathit{psec})^\vee$.

Similarly, *none* of the disjuncts of each negated disjunction $\neg\,[\,\bigvee_{\Psi \in C} \Psi\,]$ of the negative knowledge $\mathit{weak}(r, \mathit{psec})^-$ is satisfied by $\mathcal{I}_r$ – otherwise, the corresponding cluster C would be in the subset $\mathcal{C}_r^*$ of those clusters, whose corresponding disjunction templates are satisfied by the original instance r and do hence *not* result in negated disjunctions. And finally, for each constant combination $\boldsymbol{c} \in \mathit{Dom}^n$, for which $\neg R(\boldsymbol{c})$ holds according to the completeness sentence of the negative knowledge $\mathit{weak}(r, \mathit{psec})^-$, the tuple $\boldsymbol{c}$ is indeed invalid in r, as each valid tuple $\boldsymbol{d} \in r$ with $\boldsymbol{d} = (d_1, \ldots, d_n)$ either

- results in the ground atom $R(\boldsymbol{d}) \in \mathit{weak}(r, \mathit{psec})^+$, if there is *no* potential secret $\Psi \in \bigcup_{C \in \mathcal{C}_r^*} C$ with $R(\boldsymbol{d}) \models_{DB} \Psi$ or
- results in a disjunct $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ of $\mathit{weak}(r, \mathit{psec})^\vee$, which is satisfied by the DB-Interpretation $\mathcal{I}_{\boldsymbol{d}}$ induced by $\boldsymbol{d}$, if there is a potential secret $\Psi \in \bigcup_{C \in \mathcal{C}_r^*} C$ with $R(\boldsymbol{d}) \models_{DB} \Psi$,

and in both of these cases the completeness sentence contains either

- the disjunct $(\,\bigwedge_{i \in \{1,\ldots,n\}} X_i \equiv d_i\,)$ or
- the disjunct $(\,\bigwedge_{i \in \{1,\ldots,n\} \text{ with } t_i \in \mathit{Dom}} X_i \equiv t_i\,)$

preventing that the last disjunct $\neg R(X_1, \ldots, X_n)$ of the completeness sentence must necessarily hold for the constant substitution, which the constant combination $\boldsymbol{d} = (d_1, \ldots, d_n)$ induces for the universally quantified variables $X_1, \ldots, X_n$ of the (partial) completeness sentence. This results in $\mathcal{I}_r \models_M \mathit{weak}(r, \mathit{psec})^-$.

Now that all basic operations are known, the overall *generic* algorithm generating an inference-proof weakened view is presented. This algorithm is generic in the sense that – beside the employed notion of admissible indistinguishabilities, which is deliberately left open as argued in Section 3.1.3 – the construction of an extended clustering is only defined in a purely declarative way in Definition 3.3. So, while the definitions of all other basic subroutines of the algorithm induce straightforward implementations, algorithms computing an extended clustering still need to be designed on the operational level.

Algorithm 3.1: Inference-Proof Weakening (Generic)

Let r be a complete database instance over a database schema $\langle R|\mathcal{A}_R|\,\emptyset\,\rangle$, let *psec* be a confidentiality policy of existentially quantified atoms and let $k^* \in \mathbb{N}$ with $k^* \geq 2$ be the minimum length each disjunction template should have. Moreover, suppose that a notion of admissible indistinguishabilities is given, which is well-defined with respect to *psec* and k^* according to Definition 3.6.

Then, a weakened view $\mathit{weak}(r, \mathit{psec})$ on r is created as follows:

- ***Stage 1*** *(independent of r): Disjoint clustering of potential secrets*
  - (i) Construct the cleaned set $\widehat{\mathit{psec}}$ based on *psec* (Def. 3.4)
  - (ii) Create an extended clustering $\mathcal{C}^*$ of $\widehat{\mathit{psec}}$ obeying the given notion of admissible indistinguishabilities and k^* (Def. 3.3)
- ***Stage 2*** *(dependent on r): Creation of weakened view*
  - (iii) 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 satisfied by $\mathcal{I}_r$
  - (iv) Create the weakened view $\mathit{weak}(r, \mathit{psec})$ on r (Def. 3.7)

The inference-proofness of a weakened view $\mathit{weak}(r, \mathit{psec})$ returned by this algorithm crucially relies on a strict isolation of the disjunctive knowledge given in $\mathit{weak}(r, \mathit{psec})^\vee$ – which aims at *not* revealing the real truth values of the disjuncts of $\mathit{weak}(r, \mathit{psec})^\vee$ to an adversary – from the definite knowledge an adversary can gain about the original database instance. This isolation follows the goal that the satisfaction or non-satisfaction of a disjunct of $\mathit{weak}(r, \mathit{psec})^\vee$ can *not* be concluded based on the satisfaction or non-satisfaction of a piece of definite knowledge the adversary is aware of.

At first, for each ground atom $R(\boldsymbol{c})$ of the positive knowledge $\mathit{weak}(r, \mathit{psec})^+$ the non-implication $R(\boldsymbol{c}) \not\models_{DB} \Psi$ holds for each (possibly additional) potential secret

$\Psi \in \bigcup_{C \in \mathcal{C}_r^*} C$ occurring in a disjunction of $weak(r, psec)^{\vee}$. As a direct consequence, an (alternative) database instance satisfying $R(\boldsymbol{c})$ does *not* necessarily need to satisfy Ψ. Similarly, the (partial) completeness sentence of the negative knowledge $weak(r, psec)^{-}$ contains the disjunct

$$\Big(\bigwedge_{i \in \{1,\ldots,n\} \text{ with } t_i \in Dom} X_i \equiv t_i \Big)$$

for each (possibly additional) potential secret $\Psi = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ of a disjunction of $weak(r, psec)^{\vee}$. Hence, an (alternative) database instance satisfying this completeness sentence can nonetheless satisfy Ψ, as the last disjunct $\neg R(X_1, \ldots, X_n)$ of this completeness sentence does *not* need to be satisfied for each constant substitution of the universally quantified variables $X_1, \ldots, X_n$ inducing a DB-Interpretation satisfying Ψ.

Even within an extended clustering $\mathcal{C}^*$ there is *no* implication relationship between each pair Ψ_1, Ψ_2 of different (possibly additional) potential secrets of $\mathcal{C}^*$ as the set of all of these clustered potential secrets is supposed to be cleaned. Hence, an (alternative) database instance satisfying the clustered potential secret Ψ_1 does *not* necessarily need to satisfy the clustered potential secret Ψ_2 and an (alternative) instance *not* satisfying Ψ_1 might nonetheless satisfy Ψ_2. As each potential secret of the extended clustering $\mathcal{C}^*$ is moreover contained in exactly one cluster of $\mathcal{C}^*$ – due to $\mathcal{C}^*$ being a disjoint clustering – the satisfaction of a disjunction of $weak(r, psec)^{\vee}$ and the non-satisfaction of a disjunction template, whose negation is in $weak(r, psec)^{-}$, do *not* influence each other.

To exemplify that this isolation within an extended clustering is of crucial importance, suppose that the clustering algorithm creates the cluster $C_1 = \{\Psi_1, \Psi_2\}$. Further, suppose that the corresponding disjunction template $\Psi_1 \vee \Psi_2$ is *not* satisfied by a considered original database instance r and hence leads to the construction of the corresponding negated disjunction $\neg[\Psi_1 \vee \Psi_2]$ providing the definite knowledge that *neither* Ψ_1 *nor* Ψ_2 is satisfied by this original instance r. If the clustering algorithm further created the cluster $C_2 = \{\Psi_3, \Psi_4\}$ with $\Psi_3 \models_{DB} \Psi_2$ (possibly because of $\Psi_2 = \Psi_3$ due to a non-disjoint clustering), whose corresponding disjunction template $\Psi_3 \vee \Psi_4$ is satisfied by the considered original instance r, the adversary would be able to reason that Ψ_3 is *not* satisfied by this instance r (cf. Lemma 3.1). As he moreover knows that the disjunction $\Psi_3 \vee \Psi_4$ is satisfied by r because of being in $weak(r, psec)^{\vee}$, he can infer that Ψ_4 must be satisfied by r, thereby violating the potential secret Ψ_4 of the cluster C_2.

Considering the isolation properties discussed above – whose purpose is to prevent the flow of definite knowledge into disjunctive knowledge – each alternative database instance, which

- satisfies all ground atoms of the positive knowledge $weak(r, psec)^+$,
- satisfies an *arbitrary* non-empty subset of disjuncts of each disjunction of the disjunctive knowledge $weak(r, psec)^\vee$ without satisfying a disjunction template, whose negated disjunction is in $weak(r, psec)^-$, and
- does *not* satisfy any other knowledge
  (and hence satisfies the completeness sentence of $weak(r, psec)^-$),

is consistent with the weakened view $weak(r, psec)$ and is therefore indistinguishable from the original instance r from an adversary's point of view. If such an alternative instance r^Ψ does further *not* satisfy a potential secret Ψ – by satisfying a corresponding non-empty subset of disjuncts of the disjunction template containing Ψ, if this disjunction template is satisfied by the original instance – this alternative instance r^Ψ also obeys this potential secret Ψ as required by the definition of inference-proofness.

3.3 Inference-Proofness of the Generic Approach

Now that the generic algorithm is formalized, the inference-proofness of this algorithm is formally verified. Similar to the proofs of other approaches to Controlled Interaction Execution (cf. [13, 15, 27, 30]), this proof is basically achieved by providing a generic method to construct alternative database instances, which are both indistinguishable from a considered original database instance from an adversary's point of view and do *not* satisfy a particular potential secret of a considered confidentiality policy. Such an alternative instance hence serves as a witness that the considered potential secret does *not* necessarily need to be satisfied from an adversary's point of view.

As the set of all those (possibly additional) potential secrets, which occur as disjuncts of disjunction templates, is supposed to be cleaned, the definition of DB-Implication – or, more precisely, non-implication – guarantees that for each disjunct of such a disjunction template there is an "exclusively satisfying" DB-Interpretation. This DB-Interpretation consists of only one tuple and satisfies this particular disjunct, but does *not* satisfy any other disjunct occurring in a (possibly different) disjunction template. Hence, within the main proof an alternative database instance satisfying a particular subset of disjuncts of a weakened view – without accidentally satisfying some other disjuncts not to be satisfied – can be constructed by uniting the tuples of the corresponding "exclusively satisfying" DB-Interpretations. For convenience, these "exclusively satisfying" DB-Interpretations are supposed to be constructed by a so-called ground operator.

Definition 3.8: Ground Operator

Consider a cleaned and finite set $\widehat{\mathcal{S}}$ of existentially quantified atoms all constructed over a predicate symbol R of arity n and a sentence $\Psi \in \widehat{\mathcal{S}}$. The *ground operator* $grnd(\Psi, \widehat{\mathcal{S}})$ deterministically returns a constant combination $\boldsymbol{c} \in Dom^n$ such that the induced DB-Interpretation $\mathcal{I}_{\boldsymbol{c}}$ with $\mathcal{I}_{\boldsymbol{c}}(R) = \{\boldsymbol{c}\}$

(i) satisfies the existentially quantified atom Ψ, i.e., $\mathcal{I}_{\boldsymbol{c}} \models_M \Psi$, and

(ii) does *not* satisfy any other existentially quantified atom $\bar{\Psi} \in \widehat{\mathcal{S}} \setminus \{\Psi\}$, i.e., $\mathcal{I}_{\boldsymbol{c}} \not\models_M \bar{\Psi}$ for each $\bar{\Psi} \in \widehat{\mathcal{S}} \setminus \{\Psi\}$.

As this ground operator will be used within a formal proof to show the inference-proofness of the (generic) weakening algorithm, it is of importance to also give a formal proof that this ground operator itself is well-defined.

Lemma 3.2: Well-Defined Ground Operator

Let $\widehat{\mathcal{S}}$ be a cleaned and finite set of existentially quantified atoms all constructed over a predicate symbol R of arity n. For each sentence $\Psi \in \widehat{\mathcal{S}}$ the ground operator $grnd(\Psi, \widehat{\mathcal{S}})$ is able to return a constant combination $\boldsymbol{c} \in Dom^n$ as required by Definition 3.8.

Proof. Consider an arbitrary existentially quantified atom $\Psi = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ of the given set $\widehat{\mathcal{S}}$. To construct the constant combination $\boldsymbol{c} = (c_1, \ldots, c_n)$ to be returned by $grnd(\Psi, \widehat{\mathcal{S}})$, set $c_i := t_i$ for each i with $t_i \in Dom$. Moreover, for each i with $t_i \in \boldsymbol{X}$ deterministically choose c_i from the subset

$$Dom \setminus \{\, \bar{t}_i \mid (\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n) \in \widehat{\mathcal{S}} \text{ and } \bar{t}_i \in Dom \text{ and } 1 \leq i \leq n \,\}$$

of constant symbols, i.e., the subset of constant symbols of Dom which is *not* contained in the active domain of $\widehat{\mathcal{S}}$. This construction is always possible as Dom is supposed to be an infinite set of constant symbols while the set $\widehat{\mathcal{S}}$ is a finite set of existentially quantified atoms each of which is of finite arity.[3]

Obviously, this construction of $\boldsymbol{c}$ guarantees that the induced DB-Interpretation $\mathcal{I}_{\boldsymbol{c}}$ with $\mathcal{I}_{\boldsymbol{c}}(R) = \{\boldsymbol{c}\}$ satisfies the existentially quantified atom Ψ, i.e., $\mathcal{I}_{\boldsymbol{c}} \models_M \Psi$.

[3] If the ground operator is applied to a cleaned confidentiality policy $\widehat{psec}$ and in practical scenarios only a *finite* set Dom of constant symbols is available as argued in Section 3.1.3, a well-defined notion of admissible indistinguishabilities guarantees the existence of at least one "fresh" constant symbol *not* contained in the active domain of $\widehat{psec}$ according to Definition 3.6.

To furthermore assure that $\mathcal{I}_{\boldsymbol{c}}$ does *not* satisfy any other existentially quantified atom of $\widehat{\mathcal{S}} \setminus \{\Psi\}$, assume that there is such a sentence $\bar{\Psi} = (\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ different from Ψ in $\widehat{\mathcal{S}} \setminus \{\Psi\}$ with $\mathcal{I}_{\boldsymbol{c}} \models_M \bar{\Psi}$. Then, for each i with $t_i \in \mathit{Dom}$ either $\bar{t}_i = c_i = t_i$ or $\bar{t}_i \in \boldsymbol{Y}$ holds and for each i with $t_i \in \boldsymbol{X}$ the term $\bar{t}_i$ must be a variable of $\boldsymbol{Y}$ because of choosing c_i from a subset of constant symbols *not* containing any constant symbol occurring in a sentence of $\widehat{\mathcal{S}}$. Hence, by applying Lemma 2.1, the implication $\Psi \models_{DB} \bar{\Psi}$ holds in contradiction to the assumption that $\widehat{\mathcal{S}}$ is a cleaned set. ♠

According to the semantics of potential secrets, a potential secret Ψ only needs to be (actively) protected, if Ψ is satisfied by the original instance r considered. If Ψ is instead *not* satisfied by r, this potential secret Ψ is already obeyed by r and there is hence no need to prevent an adversary from knowing the real truth value of this potential secret [8, 10, 11, 13]. Thus, in the following Theorem 3.1 the existence of a certain number of different "secure" alternative instances protecting a potential secret Ψ is only required, if Ψ is actually satisfied by r.

Note that this notion of inference-proofness, which guarantees the existence of a certain number of different "secure" alternative instances for each satisfied policy element, clearly strengthens the notion of inference-proofness used in prior work on "Controlled Interaction Execution", which guarantees the existence of *only one* "secure" alternative instance for each policy element [8, 10, 11, 13, 15].

Theorem 3.1: Inference-Proofness of the Generic Approach

Let r be a complete database instance over a database schema $\langle R|\mathcal{A}_R|\,\emptyset\,\rangle$, let *psec* be a confidentiality policy of existentially quantified atoms and let $k^* \in \mathbb{N}$ with $k^* \geq 2$ be the minimum length each disjunction template should have. Moreover, suppose that a notion of admissible indistinguishabilities is given, which is well-defined with respect to *psec* and k^* according to Definition 3.6.

Algorithm 3.1 then creates a weakened view $\mathit{weak}(r, \mathit{psec})$ on the database instance r, which is inference-proof in the sense that for each potential secret $\Psi \in \mathit{psec}$ with $\mathcal{I}_r \models_M \Psi$ the existence of at least $2^{k^*-1} - 1$ pairwise different complete alternative instances r_i^{Ψ} (with $1 \leq i \leq 2^{k^*-1} - 1$) over schema $\langle R|\mathcal{A}_R|\,\emptyset\,\rangle$ is guaranteed. Each of these alternative instances r_i^{Ψ}

(i) obeys the potential secret Ψ, i.e., $\mathcal{I}_{r_i^{\Psi}} \not\models_M \Psi$, and

(ii) the corresponding weakened view $\mathit{weak}(r_i^{\Psi}, \mathit{psec})$ is indistinguishable from $\mathit{weak}(r, \mathit{psec})$, i.e., $\mathit{weak}(r_i^{\Psi}, \mathit{psec}) = \mathit{weak}(r, \mathit{psec})$.

Proof. Consider an arbitrary potential secret $\tilde{\Psi} \in psec$, which is satisfied by the original database instance r, i.e., $\mathcal{I}_r \models_M \tilde{\Psi}$. By construction of the cleaned confidentiality policy $\widehat{psec}$ there is a (weakest) potential secret $\hat{\Psi} \in \widehat{psec}$ such that the implication $\tilde{\Psi} \models_{DB} \hat{\Psi}$ holds. Then, the potential secret $\hat{\Psi}$ is also satisfied by the considered original instance r, i.e., $\mathcal{I}_r \models_M \hat{\Psi}$, as a direct consequence of $\mathcal{I}_r \models_M \tilde{\Psi}$ and $\tilde{\Psi} \models_{DB} \hat{\Psi}$. Moreover, suppose that Stage 1 of Algorithm 3.1 generated an extended clustering $\mathcal{C}^*$ according to Definition 3.3 consisting of pairwise disjoint clusters each of which is of a minimum size of k^*. The existence of such an extended clustering is guaranteed according to Definition 3.6, as the given notion of admissible indistinguishabilities is supposed to be well-defined. Further, assume that $\hat{\Psi}$ is in the cluster $\hat{C} \in \mathcal{C}^*$.

Now, it is shown that at least $2^{k^*-1} - 1$ pairwise different complete alternative instances obeying the potential secret $\hat{\Psi}$ (and hence also $\tilde{\Psi}$, as discussed later) can be constructed based on the ground operator introduced in Definition 3.8. For that purpose, consider the set

$$\widehat{psec}^* := \bigcup_{C \in \mathcal{C}^*} C$$

of all (possibly additional) potential secrets occurring in a cluster of the extended clustering $\mathcal{C}^*$ as well as the subset

$$\widehat{psec}^+ := \{\, \Psi \in \widehat{psec}^* \mid \mathcal{I}_r \models_M \Psi \,\}$$

of those (possibly additional) potential secrets, which are satisfied by the original instance r. Then, assuming that $\mathfrak{P}(\mathcal{S})$ denominates the power set of a set $\mathcal{S}$, for *each* possible subset

$$\mathscr{C}_i \in \mathfrak{P}(\hat{C} \setminus \{\hat{\Psi}\}) \;\text{ with }\; \mathscr{C}_i \neq \emptyset$$

a complete alternative instance $r_i^{\hat{\Psi}}$ protecting $\hat{\Psi}$ can be constructed by adding

(i) the tuple $\boldsymbol{c}$ for each ground atom $R(\boldsymbol{c}) \in weak(r, psec)^+$,

(ii) the tuple $grnd(\Psi, \widehat{psec}^*)$ for each
(possibly additional) potential secret $\Psi \in (\widehat{psec}^+ \setminus \hat{C})$ and

(iii) the tuple $grnd(\Psi, \widehat{psec}^*)$ for each
(possibly additional) potential secret $\Psi \in \mathscr{C}_i$

to this initially empty database instance $r_i^{\hat{\Psi}}$. This construction instantiates the ground operator only with valid arguments, as the set $\widehat{psec}^*$ contains all non-additional potential secrets of the cleaned and finite confidentiality policy $\widehat{psec}$ and each of the finitely many additional potential secrets extending $\widehat{psec}$ such that the union of all non-additional potential secrets and all additional potential secrets is

again a cleaned (cf. Definition 3.5) and finite set of existentially quantified atoms. Further, note that the sequence in which these tuples are added to $r_i^{\hat{\Psi}}$ is *not* of importance, as the ground operator only depends on a potential secret $\Psi \in \widehat{psec}^*$ and on the cleaned and finite set $\widehat{psec}^*$ of all (possibly additional) potential secrets, which remains unchanged during the construction of $r_i^{\hat{\Psi}}$.

Two complete alternative instances $r_i^{\hat{\Psi}}$ and $r_j^{\hat{\Psi}}$ are different, if their corresponding sets $\mathscr{C}_i$ and $\mathscr{C}_j$ in the form of non-empty subsets of $\hat{C} \setminus \{\hat{\Psi}\}$ are different, i.e., there is – without loss of generality – a (possibly additional) potential secret $\Psi \in \mathscr{C}_i$, which is *not* in $\mathscr{C}_j$. Then, $r_i^{\hat{\Psi}}$ contains the tuple $grnd(\Psi, \widehat{psec}^*)$ according to construction rule (iii), but this tuple is *not* contained in $r_j^{\hat{\Psi}}$. First of all, there is *no* tuple $\boldsymbol{c} \in r_j^{\hat{\Psi}}$, which is added for a ground atom $R(\boldsymbol{c}) \in weak(r, psec)^+$ according to construction rule (i) and which is equal to $grnd(\Psi, \widehat{psec}^*)$, because the non-implication $R(\boldsymbol{c}) \not\models_{DB} \Psi$ holds: if Ψ is in a cluster of $\mathcal{C}_r^*$, this non-implication is guaranteed by the construction of $weak(r, psec)^+$ according to Definition 3.7; if Ψ is *not* in a cluster of $\mathcal{C}_r^*$, the DB-Interpretation $\mathcal{I}_r$ induced by the original instance r satisfies $R(\boldsymbol{c})$ because of $\boldsymbol{c} \in r$, but does *not* satisfy Ψ as a direct consequence of the construction of $\mathcal{C}_r^*$, thereby establishing the non-implication.

Moreover, the considered tuple $grnd(\Psi, \widehat{psec}^*)$ of $r_i^{\hat{\Psi}}$ is *not* added to $r_j^{\hat{\Psi}}$ by the remaining construction rules, either: the potential secret Ψ is neither – in case of construction rule (ii) – contained in the set $\widehat{psec}^+ \setminus \hat{C}$ because of $\Psi \in \mathscr{C}_i \subseteq \hat{C}$ nor – in case of construction rule (iii) – in the set $\mathscr{C}_j$ because of the assumption $\Psi \notin \mathscr{C}_j$ and each other tuple $grnd(\bar{\Psi}, \widehat{psec}^*)$ with $\bar{\Psi} \neq \Psi$, which is added to $r_j^{\hat{\Psi}}$ for a (possibly additional) potential secret of the set $\widehat{psec}^+ \setminus \hat{C}$ or the set $\mathscr{C}_j$, is *not* equal to $grnd(\Psi, \widehat{psec}^*)$ by the construction of the ground operator according to Lemma 3.2. As hence each non-empty subset $\mathscr{C}_i \in \mathfrak{P}(\hat{C} \setminus \{\hat{\Psi}\})$ induces an alternative instance different from the alternative instances induced by all other non-empty subsets $\mathscr{C}_j \in \mathfrak{P}(\hat{C} \setminus \{\hat{\Psi}\})$ with $\mathscr{C}_i \neq \mathscr{C}_j$ and as the power set $\mathfrak{P}(\hat{C} \setminus \{\hat{\Psi}\})$ contains a total number of $2^{|\hat{C} \setminus \{\hat{\Psi}\}|} - 1$ different non-empty subsets of potential secrets of $\hat{C} \setminus \{\hat{\Psi}\}$, there is also a minimum number of

$$2^{|\hat{C} \setminus \{\hat{\Psi}\}|} - 1 = 2^{|\hat{C}|-1} - 1 \geq 2^{k^*-1} - 1$$

different complete alternative instances. Note that the existence of at least one alternative instance is always guaranteed due to $k^* \geq 2$.

In the following consider an arbitrary complete alternative instance $r_i^{\hat{\Psi}}$ constructed on the basis of a corresponding non-empty subset $\mathscr{C}_i \in \mathfrak{P}(\hat{C} \setminus \{\hat{\Psi}\})$. By the construction of $r_i^{\hat{\Psi}}$, the induced DB-Interpretation $\mathcal{I}_{r_i^{\hat{\Psi}}}$ does *not* satisfy the considered

potential secret $\hat{\Psi}$, i.e., there is *no* single tuple $\boldsymbol{c} \in r_i^{\hat{\Psi}}$ inducing a DB-Interpretation satisfying $\hat{\Psi}$. First of all, each tuple $\boldsymbol{c} \in r_i^{\hat{\Psi}}$ stemming from a ground atom $R(\boldsymbol{c}) \in \mathit{weak}(r, \mathit{psec})^+$ according to construction rule (i) does *not* induce such a DB-Interpretation, as $\hat{\Psi}$ is supposed to be in a cluster of $\mathcal{C}_r^*$ due to $\mathcal{I}_r \models_M \hat{\Psi}$ and as hence the non-implication $R(\boldsymbol{c}) \not\models_{DB} \hat{\Psi}$ holds by construction of $\mathit{weak}(r, \mathit{psec})^+$ according to Definition 3.7.

Moreover, each tuple $\mathit{grnd}(\Psi, \widehat{\mathit{psec}}^*) \in r_i^{\hat{\Psi}}$ constructed for a (possibly additional) potential secret $\Psi \in \widehat{\mathit{psec}}^*$ by the remaining construction rules does *not* induce such a DB-Interpretation satisfying $\hat{\Psi}$, either: the tuple $\mathit{grnd}(\hat{\Psi}, \widehat{\mathit{psec}}^*)$ is *not* added to $r_i^{\hat{\Psi}}$, as the potential secret $\hat{\Psi}$ is neither – in case of construction rule (ii) – contained in $\widehat{\mathit{psec}}^+ \setminus \hat{C}$ because of $\hat{\Psi} \in \hat{C}$ nor – in case of construction rule (iii) – in the subset $\mathscr{C}_i \in \mathfrak{P}(\hat{C} \setminus \{\hat{\Psi}\})$ and each other tuple $\mathit{grnd}(\Psi, \widehat{\mathit{psec}}^*)$ added to $r_i^{\hat{\Psi}}$ for a (possibly additional) potential secret $\Psi \in \widehat{\mathit{psec}}^*$ with $\Psi \neq \hat{\Psi}$ does *not* induce a DB-Interpretation satisfying $\hat{\Psi}$ by construction of the ground operator according to Lemma 3.2. So, by $\mathcal{I}_{r_i^{\hat{\Psi}}} \not\models_M \hat{\Psi}$ and $\tilde{\Psi} \models_{DB} \hat{\Psi}$ and by further applying Lemma 3.1, the constructed alternative instance $r_i^{\hat{\Psi}}$ does *not* satisfy the potential secret $\tilde{\Psi}$ to be protected, i.e., $\mathcal{I}_{r_i^{\hat{\Psi}}} \not\models_M \tilde{\Psi}$.

As the potential secret $\hat{\Psi}$ of the cluster $\hat{C} \in \mathcal{C}^*$ is supposed to be satisfied by the original instance r, the disjunction template $\bigvee_{\Psi \in \hat{C}} \Psi$ corresponding to this cluster $\hat{C}$ is satisfied by r, too. This disjunction template $\bigvee_{\Psi \in \hat{C}} \Psi$ is also satisfied by the considered complete alternative instance $r_i^{\hat{\Psi}}$, i.e., $\mathcal{I}_{r_i^{\hat{\Psi}}} \models_M \bigvee_{\Psi \in \hat{C}} \Psi$, as $\mathscr{C}_i$ is supposed to be a non-empty subset of $\hat{C} \setminus \{\hat{\Psi}\}$ and for each (possibly additional) potential secret $\Psi \in \mathscr{C}_i$ the tuple $\mathit{grnd}(\Psi, \widehat{\mathit{psec}}^*)$ is added to $r_i^{\hat{\Psi}}$ by construction rule (iii). This guarantees that the alternative instance $r_i^{\hat{\Psi}}$ satisfies each potential secret $\Psi \in \mathscr{C}_i \subseteq \hat{C}$ – and hence also the disjunction template $\bigvee_{\Psi \in \hat{C}} \Psi$ – due to the construction of the ground operator according to Lemma 3.2.

For each other cluster $C \in \mathcal{C}^*$ with $C \neq \hat{C}$ a (possibly additional) potential secret $\Psi \in C$ is satisfied by the considered alternative instance $r_i^{\hat{\Psi}}$ *if and only if* it is satisfied by the given original instance r. If such a potential secret $\Psi \in C$ is satisfied by the original instance r, i.e., $\mathcal{I}_r \models_M \Psi$, this potential secret Ψ is contained in the subset $\widehat{\mathit{psec}}^+$ of those (possibly additional) potential secrets of $\widehat{\mathit{psec}}^*$, which are satisfied by r. As the clusters C and $\hat{C}$ of the extended clustering $\mathcal{C}^*$ are supposed to be disjoint, the considered potential secret Ψ of C can *not* be in the cluster $\hat{C}$ and is hence contained in the set $\widehat{\mathit{psec}}^+ \setminus \hat{C}$. Consequently, the tuple $\mathit{grnd}(\Psi, \widehat{\mathit{psec}}^*)$ is added to the alternative instance $r_i^{\hat{\Psi}}$ by construction rule (ii) and the considered potential secret Ψ is thus satisfied by $r_i^{\hat{\Psi}}$ due to the

construction of the ground operator according to Lemma 3.2.

If a (possibly additional) potential secret Ψ of the considered cluster C is *not* satisfied by the original instance r, i.e., $\mathcal{I}_r \not\models_M \Psi$, the alternative instance $r_i^{\hat{\Psi}}$ does *not* satisfy this potential secret Ψ, either, as there is *no* single tuple $\boldsymbol{c} \in r_i^{\hat{\Psi}}$ inducing a DB-Interpretation satisfying Ψ. Each tuple $\boldsymbol{c} \in r_i^{\hat{\Psi}}$ stemming from a ground atom $R(\boldsymbol{c}) \in \mathit{weak}(r, \mathit{psec})^+$ according to construction rule (i) does *not* induce such a DB-Interpretation: otherwise, the original instance r, which satisfies $R(\boldsymbol{c})$ because of $\boldsymbol{c} \in r$, would also satisfy the considered potential secret Ψ in contradiction to the assumption. Moreover, each tuple $\mathit{grnd}(\bar{\Psi}, \widehat{\mathit{psec}}^*) \in r_i^{\hat{\Psi}}$ constructed for a (possibly additional) potential secret $\bar{\Psi} \in \widehat{\mathit{psec}}^*$ by the remaining construction rules (ii) and (iii) does *not* induce such a DB-Interpretation, either: the tuple $\mathit{grnd}(\Psi, \widehat{\mathit{psec}}^*)$ is *not* added to $r_i^{\hat{\Psi}}$, as the construction of $\widehat{\mathit{psec}}^+$ guarantees $\Psi \notin \widehat{\mathit{psec}}^+$ due to the assumption $\mathcal{I}_r \not\models_M \Psi$ and by further ensuring $\Psi \notin \mathscr{C}_i$ due to the disjoint clustering. Each other tuple $\mathit{grnd}(\bar{\Psi}, \widehat{\mathit{psec}}^*)$ added to $r_i^{\hat{\Psi}}$ with $\bar{\Psi} \neq \Psi$ does *not* induce a DB-Interpretation satisfying Ψ by the construction of the ground operator according to Lemma 3.2.

Summarizing the results above, each (possibly additional) potential secret Ψ of a cluster $C \in \mathcal{C}^*$ with $C \neq \hat{C}$ is satisfied by the considered alternative instance $r_i^{\hat{\Psi}}$, *if and only if* it is satisfied by the given original instance r. The disjunction template $\bigvee_{\Psi \in \hat{C}} \Psi$ corresponding to the cluster $\hat{C}$, which is supposed to contain the potential secret $\hat{\Psi}$, is moreover satisfied by both the original instance r and the alternative instance $r_i^{\hat{\Psi}}$. Hence, for *each* cluster $C \in \mathcal{C}^*$ (including $C = \hat{C}$) the corresponding disjunction template $\bigvee_{\Psi \in C} \Psi$ is satisfied by the considered alternative instance $r_i^{\hat{\Psi}}$, *if and only if* it is satisfied by original instance r. This immediately implies $\mathcal{C}^*_{r_i^{\hat{\Psi}}} = \mathcal{C}^*_r$ and hence also $\mathit{weak}(r_i^{\hat{\Psi}}, \mathit{psec})^\vee = \mathit{weak}(r, \mathit{psec})^\vee$.

For each ground atom $R(\boldsymbol{c}) \in \mathit{weak}(r, \mathit{psec})^+$ the tuple $\boldsymbol{c}$ is contained in $r_i^{\hat{\Psi}}$ due to construction rule (i) and – by the construction of $\mathit{weak}(r, \mathit{psec})^+$ according to Definition 3.7 – the non-implication $R(\boldsymbol{c}) \not\models_{DB} \Psi$ is guaranteed to hold for each (possibly additional) potential secret Ψ of a cluster of $\mathcal{C}^*_r$. Again considering $\mathcal{C}^*_{r_i^{\hat{\Psi}}} = \mathcal{C}^*_r$, this non-implication $R(\boldsymbol{c}) \not\models_{DB} \Psi$ also holds for each (possibly additional) potential secret Ψ of a cluster of $\mathcal{C}^*_{r_i^{\hat{\Psi}}}$ and hence this tuple $\boldsymbol{c}$ of the alternative instance $r_i^{\hat{\Psi}}$ is also contained in $\mathit{weak}(r_i^{\hat{\Psi}}, \mathit{psec})^+$ according to Definition 3.7. For each other tuple $\mathit{grnd}(\Psi, \widehat{\mathit{psec}}^*)$ added to $r_i^{\hat{\Psi}}$ for a (possibly additional) potential secret $\Psi \in \widehat{\mathit{psec}}^*$ by the remaining construction rules (ii) and (iii), the construction of the ground operator guarantees the validity of the implication $R(\mathit{grnd}(\Psi, \widehat{\mathit{psec}}^*)) \models_{DB} \Psi$ and hence – additionally considering that Ψ is in a

cluster of $\mathcal{C}^*_{r_i^{\hat{\Psi}}}$ due to $grnd(\Psi, \widehat{psec}^*) \in r_i^{\hat{\Psi}}$ – the tuple $grnd(\Psi, \widehat{psec}^*)$ is *not* qualified for being in $weak(r_i^{\hat{\Psi}}, psec)^+$ according to Definition 3.7. This results in $weak(r_i^{\hat{\Psi}}, psec)^+ = weak(r, psec)^+$.

Reconsidering the construction of the negative knowledge of a weakened view according to Definition 3.7 – which consists of a sequence of negated disjunctions and a (partial) completeness sentence – the validity of both

- $weak(r_i^{\hat{\Psi}}, psec)^+ = weak(r, psec)^+$ and
- $weak(r_i^{\hat{\Psi}}, psec)^\vee = weak(r, psec)^\vee$ as well as
- $\mathcal{C}^* \setminus \mathcal{C}^*_{r_i^{\hat{\Psi}}} = \mathcal{C}^* \setminus \mathcal{C}^*_r$ (due to $\mathcal{C}^*_{r_i^{\hat{\Psi}}} = \mathcal{C}^*_r$)

immediately leads to $weak(r_i^{\hat{\Psi}}, psec)^- = weak(r, psec)^-$. Hence, indistinguishability is achieved because of $weak(r_i^{\hat{\Psi}}, psec) = weak(r, psec)$, provided that the sentences of both of these sequences are arranged in the same order. ♠

As mentioned above, Theorem 3.1 only requires the existence of a certain minimum number of different "secure" alternative instances for potential secrets, which are satisfied by the original instance r. But if instead a potential secret $\Psi \in psec$ with $\mathcal{I}_r \not\models_M \Psi$ is considered, the existence of *at least one* complete alternative instance r^Ψ protecting Ψ can nonetheless be guaranteed: in this case the given original instance r can serve as an alternative instance r^Ψ, i.e., $r^\Psi := r$. This construction of r^Ψ directly implies $\mathcal{I}_{r^\Psi} \not\models_M \Psi$ and consequently the constructed alternative instance r^Ψ obeys Ψ. Furthermore, the property of indistinguishability, i.e., $weak(r^\Psi, psec) = weak(r, psec)$, is a direct consequence of $r^\Psi = r$.

To exemplify that it is indeed *not* always possible to construct more than one alternative instance protecting a particular potential secret *not* satisfied by a given original instance, consider

- the original instance $r = \emptyset$ and
- the (cleaned) policy $psec = \{ R(\boldsymbol{c}_1), R(\boldsymbol{c}_2) \}$ with $\boldsymbol{c}_1, \boldsymbol{c}_2 \in Dom^n$

and suppose that Stage 1 of Algorithm 3.1 creates the extended clustering $\mathcal{C}^*$ consisting of the single cluster

$$C = \{ R(\boldsymbol{c}_1), R(\boldsymbol{c}_2) \} \ .$$

Then, Stage 2 of Algorithm 3.1 creates a weakened view $weak(r, psec)$ containing the sequences

- $weak(r, psec)^{+} = \emptyset$ and
- $weak(r, psec)^{\vee} = \emptyset$.

Now, suppose that the potential secret $\Psi := R(\boldsymbol{c}_1)$ of $psec$ is considered to be protected. As described above, it is possible to construct the alternative instance $r_1^{\Psi} := r = \emptyset$, which obeys Ψ and which is indistinguishable from r as well. To create an alternative instance r_2^{Ψ} also obeying $\Psi = R(\boldsymbol{c}_1)$ but being different from $r_1^{\Psi} = \emptyset$, this instance r_2^{Ψ} must contain at least one tuple $\boldsymbol{c} \in Dom^n$ with $\boldsymbol{c} \neq \boldsymbol{c}_1$. If $\boldsymbol{c} = \boldsymbol{c}_2$, the corresponding weakened view $weak(r_2^{\Psi}, psec)$ contains the disjunction $R(\boldsymbol{c}_1) \vee R(\boldsymbol{c}_2)$ because of $\mathcal{I}_{r_2^{\Psi}} \models_M R(\boldsymbol{c}_2)$ and hence also $\mathcal{I}_{r_2^{\Psi}} \models_M \bigvee_{\Psi \in C} \Psi$. Thus, the property of indistinguishability is violated due to $weak(r_2^{\Psi}, psec)^{\vee} \neq weak(r, psec)^{\vee}$. Otherwise, if $\boldsymbol{c} \neq \boldsymbol{c}_2$, the corresponding weakened view $weak(r_2^{\Psi}, psec)$ contains the ground atom $R(\boldsymbol{c})$ because $R(\boldsymbol{c})$ does *not* imply any of the policy elements of $psec$ and thus the property of indistinguishability is violated due to $weak(r_2^{\Psi}, psec)^{+} \neq weak(r, psec)^{+}$.

3.4 Complexity of the Generic Weakening Approach

As already outlined in Section 1.3.2, the construction of weakened views is related to the well-known approaches of k-anonymization [47, 79, 86]. These approaches aim at preventing the re-identification of individuals on the basis of so-called quasi-identifiers, which describe some of the individuals' properties, by generalizing the values of these quasi-identifiers to such an extent that each individual can *not* be distinguished from (at least) $k-1$ other individuals on the basis of these quasi-identifiers. Considering only maximum generalizations of values in the form of complete suppressions and aiming at the construction of k-anonymized database instances with only a minimum number of suppressed values, this (optimization) problem is known to be NP-hard according to [3, 39] when choosing $k \geq 3$. For $k = 2$ this problem is instead known to be solvable in polynomial time.

Similarly, the developed generic weakening approach aims at the construction of disjunctions of potential secrets and such a disjunction of length k does *not* reveal which (subset) of its disjuncts is satisfied by a considered original database instance. An adversary can hence *not* distinguish, whether a certain potential secret of this disjunction or one of the $k-1$ other potential secrets of this disjunction is satisfied by the original instance. Moreover, in both of these approaches generalizations can *not* be introduced arbitrarily: k-anonymization requires that these generalizations achieve indistinguishability within subgroups of individuals violating the required indistinguishability property and the generic weakening approach

requires each disjunction template to be admissible according to some notion of admissible indistinguishabilities.

These similarities between k-anonymization and the generic weakening approach – in combination with the above mentioned results about the complexity of k-anonymization – suggest that the deterministic computation of weakened views with disjunctions of a minimum length of 3 (always guaranteeing more than one alternative instance) is in general *not* possible in polynomial time, as long as $\mathsf{NP} \neq \mathsf{P}$ is supposed to hold [48].

In the following, a formal analysis of the complexity of the generic weakening algorithm is provided. For that purpose, its subroutine for computing an extended clustering is considered under the commonly used optimization goal that only a minimum number of additional potential secrets – each of which introduces additional knowledge to be kept confidential and hence reduces availability – should be employed for the construction of an extended clustering (cf. Section 3.1.1).

As the complexity class NP contains only decision problems [48, 65, 90], a decision variant of the extended clustering problem under the above mentioned optimization goal is now developed to show that this decision variant is NP-complete. To capture the optimization goal of minimizing the employed number of additional potential secrets within this decision variant, a valid extended clustering is only accepted by the decision variant of the extended clustering problem, if it – similar to a decision variant for the well-known "Traveling Salesman"-problem (cf. [65]) – does *not* exceed a certain upper bound of costs.

Definition 3.9: Decision Variant of Extended Clustering

Let *psec* be a confidentiality policy and suppose that a notion of admissible indistinguishabilities is given. Further, suppose that $k^* \in \mathbb{N}$ with $k^* \geq 2$ is the minimum size each (extended) cluster should have and let $c \in \mathbb{N}_0$ be the maximum number of additional potential secrets, which are allowed to be in an extended clustering.

Then, the policy *psec*, the given notion of admissible indistinguishabilities and the values k^* and c form a *valid input* for the *decision variant* of the extended clustering problem, if

(i) there is an extended clustering $\mathcal{C}^*$ of the confidentiality policy *psec* obeying the given notion of admissible indistinguishabilities and k^* according to Definition 3.3 such that

(ii) this extended clustering $\mathcal{C}^*$ contains at most c additional potential secrets *not* stemming from *psec*, i.e.,

$$\sum_{C \in \mathcal{C}^*} |\, C \setminus psec \,| \leq c \ .$$

Otherwise, the policy *psec*, the given notion of admissible indistinguishabilities and the values k^* and c form a *non-valid input* for this decision problem.

As usual, the NP-completeness of this decision variant of the extended clustering problem is proved by showing that this decision problem is in NP – i.e., checking the validity of an input instance must be possible in polynomial time in the size of this input instance – and by reducing another "reference" decision problem, which is already known to be NP-complete, to this decision variant of the extended clustering problem [65, 90]. The goal of this reduction is to prove that each (original) input instance for the reference decision problem can be transformed in polynomial time in the size of this (original) input instance into a (transformed) input instance for the decision variant of the extended clustering problem such that the original input is valid for the reference decision problem, *if and only if* the transformed input is valid for the decision variant of the extended clustering problem. Then, it is shown that the decision variant of the extended clustering problem is generally *not* easier to solve than the NP-complete reference problem.

In the remainder of this complexity analysis of the decision variant of the extended clustering problem, the NP-complete "Exact Cover by 3-Sets"-problem [58] serves as such a reference decision problem.

Definition 3.10: Exact Cover by 3-Sets

Let $\mathcal{X}$ be a set of elements with $|\mathcal{X}| = 3p$ for an arbitrary $p \in \mathbb{N} \setminus \{0\}$ and let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be a collection of 3-element subsets of $\mathcal{X}$, i.e., $S_i \subseteq \mathcal{X}$ with $|S_i| = 3$ for each $i \in \{1, \ldots, m\}$.

The sets $\mathcal{X}$ and $\mathcal{S}$ form a *valid input* for the "Exact Cover by 3-Sets"-problem, if there is an *exact cover*, i.e., there is a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that

(i) for each element $x \in \mathcal{X}$ the subcollection $\mathcal{S}'$ contains a subset $S_j \in \mathcal{S}'$ with $x \in S_j$ and

(ii) $S_i \cap S_j = \emptyset$ holds for all pairs of different subsets $S_i, S_j \in \mathcal{S}'$ with $i \neq j$.

Otherwise, the sets $\mathcal{X}$ and $\mathcal{S}$ form a *non-valid input* for this decision problem.

In the following, the proof that the decision variant of the extended clustering problem is NP-complete is established under the assumption that the admissibility of a cluster according to a considered notion of admissible indistinguishabilities can be decided efficiently in polynomial time. Otherwise, this decision variant can *not* be solved efficiently anyway because of instantiating it with an input in the form of a non-efficient subroutine. Similarly, it is assumed that the requirements additional potential secrets have to satisfy according to condition (v) of Definition 3.3 can be checked efficiently in polynomial time, as this requirement is (deliberately) left generic and can hence be seen as an input-like subroutine instantiating the extended clustering problem.

Theorem 3.2: Complexity of Extended Clustering

Consider a decision variant of the extended clustering problem such that

- the admissibility of a cluster according to a considered notion of admissible indistinguishabilities can be decided in polynomial time in the size of the other inputs *psec* and k^* and
- the validity of the requirements additional potential secrets have to satisfy according to condition (v) of Definition 3.3 can be checked in polynomial time in the size of the inputs *psec* and k^*.

This decision variant of the extended clustering problem is NP-complete, i.e.,

(i) this decision problem is in NP and

(ii) an arbitrary input for the NP-complete "Exact Cover by 3-Sets"-problem can be transformed into an input for the considered decision variant of the extended clustering problem in polynomial time in the size of the input for the "Exact Cover by 3-Sets"-problem such that

(iii) this transformed input is valid for the decision variant of the extended clustering problem, *if and only if* the (original) input is valid for the "Exact Cover by 3-Sets"-problem.

Proof. It is easy to verify that the decision variant of the extended clustering problem is in NP. Reconsidering the assumptions that

- the admissibility of a cluster according to a considered notion of admissible indistinguishabilities can be decided in polynomial time in the size of the other inputs *psec* and k^* and

- the validity of the requirements additional potential secrets have to satisfy according to condition (v) of Definition 3.3 can be checked in polynomial time in the size of the inputs *psec* and k^*,

it can obviously be decided in polynomial time in the size of the remaining inputs *psec* and k^*, whether an extended clustering $\mathcal{C}^*$ complies with the requirements given in Definition 3.3.

Moreover, counting the number of additional potential secrets occurring within an extended clustering $\mathcal{C}^*$ and comparing this result with the upper bound c is also possible in polynomial time in the size of the inputs *psec*, k^* and c, as the number of clusters of an extended clustering is limited to $|psec|$ because of the requirement that each extended cluster must contain at least one non-additional potential secret. Further – in case that (unnecessarily) huge clusters are constructed with the help of additional potential – the counting of additional potential secrets can be aborted as soon as the upper bound of c is exceeded.

Now, consider an input for the "Exact Cover by 3-Sets"-problem consisting of an arbitrary set $\mathcal{X}$ of elements with $|\mathcal{X}| = 3p$ for an arbitrary $p \in \mathbb{N} \setminus \{0\}$ and an arbitrary collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ of 3-element subsets of $\mathcal{X}$. Under the further supposition that there is an injective function δ mapping each element of $\mathcal{X}$ to a potential secret in the form of a ground atom, the considered input of the "Exact Cover by 3-Sets"-problem can be transformed into an input for the decision variant of the extended clustering problem as follows:

- $psec := \{\, \delta(x) \mid x \in \mathcal{X} \,\}$ is the confidentiality policy,
- the notion of admissible indistinguishabilities is constructed such that it induces the set

$$\{\, \{\delta(x_1), \delta(x_2), \delta(x_3)\} \mid \{x_1, x_2, x_3\} \in \mathcal{S} \,\}$$

  of admissible clusters for the (non-additional) potential secrets of *psec* and further allows that each potential secret of *psec* can be grouped together with arbitrary additional potential secrets *not* occurring in *psec*,

- $k^* := 3$ is the minimum size each (extended) cluster should have,
- $c := 0$ is the maximum number of additional potential secrets and
- *no* further requirements are set up for the construction of additional potential secrets, i.e., condition (v) of Definition 3.3 is supposed to be always satisfied.

Note that such an injective function δ can always be constructed, as the set $\mathcal{X}$ is supposed to contain only a *finite* number of elements, while the infinite domain *Dom* allows for the construction of an *infinite* number of potential secrets in the form of ground atoms.

Moreover, the considered notion of admissible indistinguishabilities can be constructed in polynomial time in the size of $\mathcal{X}$: the policy *psec* is of the same cardinality as $\mathcal{X}$ and the set of admissible clusters induced for *psec* by this notion of admissible indistinguishabilities is of the same cardinality as $\mathcal{S}$. As this set $\mathcal{S}$ is supposed to contain 3-element subsets of $\mathcal{X}$, the number of induced admissible clusters is hence limited to

$$\binom{|\mathcal{X}|}{3} = \frac{|\mathcal{X}|!}{3! \cdot (|\mathcal{X}| - 3)!} = \frac{|\mathcal{X}| \cdot (|\mathcal{X}| - 1) \cdot (|\mathcal{X}| - 2)}{6} < |\mathcal{X}|^3$$

and this notion of admissible indistinguishabilities can accordingly be constructed in polynomial time in the size of $\mathcal{X}$ by exhaustively listing all admissible clusters over *psec*. Considering also the construction of the remaining inputs *psec*, k^* and c for the decision variant of the extended clustering problem, this transformation of the input for the "Exact Cover by 3-Sets"-problem obviously takes only time polynomial in the size of $\mathcal{X}$ and the size of $\mathcal{S}$.

In the remainder of this proof it is shown that $\mathcal{X}$ and $\mathcal{S}$ form a valid input for the "Exact Cover by 3-Sets"-problem, *if and only if* the transformed inputs in the form of the policy *psec*, the notion of admissible indistinguishabilities and the values k^* and c form a valid input for the decision variant of the extended clustering problem.

To prove the *if-part* of the equivalence, suppose that $\mathcal{X}$ and $\mathcal{S}$ form a valid input for the "Exact Cover by 3-Sets"-problem. Hence, there is a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that each element $x \in \mathcal{X}$ is in *exactly one* set of the subcollection $\mathcal{S}'$. Now, construct the extended clustering $\mathcal{C}^*$ by setting

$$\mathcal{C}^* := \{\, \{\delta(x_1), \delta(x_2), \delta(x_3)\} \mid \{x_1, x_2, x_3\} \in \mathcal{S}' \,\} \ .$$

Obviously, each potential secret $\delta(x) \in$ *psec* occurs in exactly one cluster of $\mathcal{C}^*$ because of x occurring in exactly one 3-element subset of the subcollection $\mathcal{S}'$ and because of the injective function δ guaranteeing that different elements $x_1, x_2 \in \mathcal{X}$ are mapped to different potential secrets $\delta(x_1), \delta(x_2) \in$ *psec*. As each cluster $\{\delta(x_1), \delta(x_2), \delta(x_3)\} \in \mathcal{C}^*$ is of size $k^* = 3$ and further admissible according to the construction of the notion of admissible indistinguishabilities, the constructed $\mathcal{C}^*$ is an extended clustering of *psec* obeying the considered notion of admissible indistinguishabilities and k^* according to Definition 3.3.

Moreover, this extended clustering does *not* contain any additional potential secrets *not* stemming from *psec* and accordingly the number of additional potential secrets occurring in $\mathcal{C}^*$ is

$$\sum_{C \in \mathcal{C}^*} |\, C \setminus psec \,| = 0 \ .$$

So, the transformed inputs in the form of the policy *psec*, the notion of admissible indistinguishabilities and the values k^* and c form a valid input for the decision variant of the extended clustering problem.

To next prove the *only-if-part* of the equivalence, suppose that the transformed inputs in the form of the policy *psec*, the notion of admissible indistinguishabilities and the values k^* and c form a valid input for the decision variant of the extended clustering problem. Consequently, there is an extended clustering $\mathcal{C}^*$ of *psec* obeying the considered notion of admissible indistinguishabilities and k^* such that further

$$\sum_{C \in \mathcal{C}^*} |\, C \setminus psec \,| \leq 0$$

holds, i.e., this extended clustering $\mathcal{C}^*$ does *not* contain any additional potential secret *not* stemming from *psec*.

Each cluster $C \in \mathcal{C}^*$ consisting only of original policy elements must be completely contained in an admissible cluster induced by the considered notion of admissible indistinguishabilities. As each of these admissible clusters is of cardinality 3, each cluster $C \in \mathcal{C}^*$ has a maximum cardinality of 3. Further considering that each cluster of $\mathcal{C}^*$ must have a minimum cardinality of $k^* := 3$, each cluster $C \in \mathcal{C}^*$ has a cardinality of exactly 3. Hence, an exact cover in the form of a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ can be constructed as

$$\mathcal{S}' := \{\, \{\delta^{-1}(\Psi_1),\, \delta^{-1}(\Psi_2),\, \delta^{-1}(\Psi_3)\} \;\mid\; \{\Psi_1,\, \Psi_2,\, \Psi_3\} \in \mathcal{C}^* \,\} \ .$$

By considering that the injective function δ guarantees that different potential secrets $\Psi_1, \Psi_2 \in psec$ have different preimages $\delta^{-1}(\Psi_1), \delta^{-1}(\Psi_2) \in \mathcal{X}$ and by further considering that each potential secret Ψ of *psec* must occur in exactly one of the pairwise disjoint clusters of the extended clustering $\mathcal{C}^*$ according to Definition 3.3, it is guaranteed that each element $\delta^{-1}(\Psi) \in \mathcal{X}$ is in a subset $S_j \in \mathcal{S}'$ and that all of these subsets contained in $\mathcal{S}'$ are pairwise disjoint. Hence, the sets $\mathcal{X}$ and $\mathcal{S}$ form a valid input for the "Exact Cover by 3-Sets"-problem. ♠

Now, it is proved that the decision variant of the extended clustering problem is NP-complete and as a consequence there is *no* deterministic algorithm solving this decision problem efficiently (in polynomial time in the size of its inputs) as long as NP $\neq$ P is supposed to hold. As an optimization problem is generally *not* easier

to solve than its corresponding decision variant [90], there is also *no* deterministic and efficient algorithm solving the above mentioned optimization variant of the extended clustering problem under the $\mathsf{NP} \neq \mathsf{P}$ hypothesis.

CHAPTER 4

An Availability-Maximizing Instantiation

Although the generic weakening algorithm developed in Chapter 3 specifies the computation of inference-proof weakened views on original database instances, this algorithm crucially relies on a so-called extended clustering of the potential secrets of a given confidentiality policy. But until now, for this extended clustering only a purely declarative definition is given, which does *not* induce a straightforward implementation on the operational level. Similarly, to keep this algorithm adaptable for different application scenarios, well-defined notions of admissible indistinguishabilities, which are needed to induce sets of admissible clusters, are only specified in a purely abstract way by giving requirements such a well-defined notion of admissible indistinguishabilities should meet.

In the following a concrete instantiation of the developed generic weakening algorithm is given on the operational level, thereby focusing on the construction of extended clusterings with clusters of size 2. This leads to a both (globally) availability-maximizing and computationally efficient instantiation of the weakening algorithm. Moreover, an example of a provably well-defined notion of admissible indistinguishabilities is introduced, which also follows the goal of (locally) maximizing availability and essentially implements the idea of protecting knowledge by generalizing certain constant values of some database tuples to wider sets of possible constant values.

4.1 An Efficient and Availability-Maximizing Clustering

As shown in Section 3.4, the construction of an extended clustering with only a minimum number of additional potential secrets and with clusters of a minimum size of $k^* \geq 3$ is NP-hard and as a direct consequence there is *no* deterministic algorithm solving this clustering problem efficiently as long as $\mathsf{NP} \neq \mathsf{P}$ is supposed to hold. To nonetheless design an *efficient* instantiation of the generic weakening algorithm, which is applicable even for large confidentiality policies, the clustering algorithm developed in the following focuses on the construction of an extended clustering with clusters of size 2.

According to the generic weakening algorithm, such an extended clustering immediately induces a set of disjunctions of length 2, which are the shortest possible non-trivial disjunctions and guarantee the existence of only one secure alternative instance for each element of the confidentiality policy. Hence, this instantiation of the generic weakening algorithm meets the minimum requirements to achieve inference-proofness in the sense of Theorem 3.1 and in turn maximizes availability with respect to the knowledge an adversary can gain about the original database instance without compromising the confidentiality policy.

4.1.1 Construction of Maximum Clusterings

Under the supposition that solely clusters of size 2 are to be constructed, it seems reasonable to further assume that each admissible cluster induced by a notion of admissible indistinguishabilities is also of size 2. Even if some of these admissible clusters are larger, each of these larger admissible clusters can be substituted by the collection of all of its 2-element subsets without affecting the resulting clustering. As an admissible cluster of size $k > 2$ has a total number of

$$\binom{k}{2} = \frac{k!}{2! \cdot (k-2)!} = \frac{k \cdot (k-1)}{2} < k^2$$

2-element subsets, this substitution of too large admissible clusters increases the total number of admissible clusters to be finally handled by the clustering algorithm only polynomially.

Obviously, an admissible cluster of size 2 corresponds to a binary relation. Hence, the set of all admissible clusters induced by a given notion of admissible indistinguishabilities can be represented by an undirected graph, which is also referred to as an indistinguishability graph.

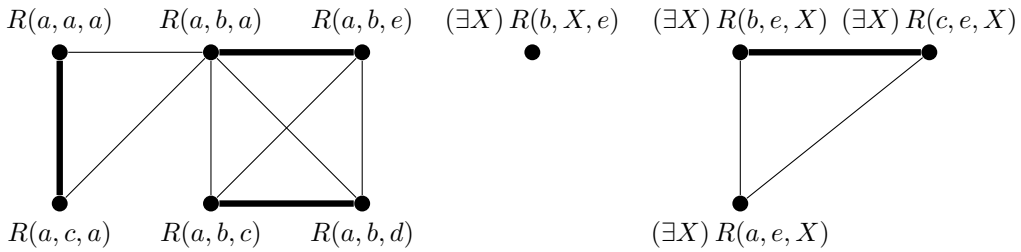


Figure 4.1: Indistinguishability graph with (bold) maximum matching

Definition 4.1: Indistinguishability Graph

Let $\widehat{psec}$ be a cleaned confidentiality policy and further suppose that $\mathcal{C}^a$ is a set of admissible clusters of size 2 over $\widehat{psec}$, which is induced by a given notion of admissible indistinguishabilities.

The *indistinguishability graph* corresponding to the set $\mathcal{C}^a$ of admissible clusters is an undirected graph $G = (V, E)$ such that

(i) $V := \widehat{psec}$ is the set of vertices of G and

(ii) each (unordered) pair $\{\Psi_1, \Psi_2\}$ with $\Psi_1, \Psi_2 \in V$ and $\Psi_1 \neq \Psi_2$ constitutes an undirected edge of E, provided that $\{\Psi_1, \Psi_2\}$ is in the set $\mathcal{C}^a$ of all admissible clusters.

An example of an indistinguishability graph for a cleaned confidentiality policy $\widehat{psec}$ is given in Figure 4.1. Each potential secret of this policy $\widehat{psec}$ is modeled as a vertex of the given graph and each admissible cluster of size 2 corresponds to an edge connecting both of the (vertices corresponding to the) policy elements contained in this admissible cluster. Thereby, each of the given admissible clusters is induced by a concrete example of a notion of admissible indistinguishabilities called "interchangeability", which is later introduced in Section 4.2.1.

Still focusing on the goal to construct an extended clustering – containing only a minimum number of additional potential secrets, thereby minimizing the knowledge to be distorted additionally – based on a given set of admissible clusters of size 2, the clustering algorithm to be developed should select a *maximum* subset of admissible clusters such that all of these selected clusters are *pairwise disjoint*. Since each of the cleaned policy elements, which is *not* contained in any of the selected admissible clusters, trivially induces a (too small) singleton cluster, the set of all selected clusters of size 2 and all singleton clusters obviously is a (*not*

necessarily extended) clustering according to Definition 3.2, which contains only a minimum number of (too small) singleton clusters.

Considering an indistinguishability graph as introduced above, such a maximum subset of selected pairwise (vertex-)disjoint admissible clusters of size 2 in the form of edges of this graph is a maximum matching on this graph. Such a maximum matching on a general, i.e., *not* necessarily bipartite, graph $G = (V, E)$ can be computed efficiently – in time $O(\sqrt{|V|} \cdot |E|)$ as demonstrated in [73, 89] – with well-known maximum matching algorithms and is defined as follows [50, 67, 77, 80].

Definition 4.2: Maximum Matching

Let $G = (V, E)$ be an undirected (not necessarily bipartite) graph without loops, i.e., $v_1 \neq v_2$ holds for each edge $\{v_1, v_2\} \in E$.

A subset $M \subseteq E$ of edges of the graph G is a *matching* on G, if

$$\{v_1, v_2\} \cap \{\bar{v}_1, \bar{v}_2\} = \emptyset$$

holds for each pair of different *matching edges* $\{v_1, v_2\}, \{\bar{v}_1, \bar{v}_2\} \in M$.

A matching M on the considered graph G is moreover

- *maximum* with respect to its cardinality, if $|M'| \leq |M|$ holds for each possible matching M' on the graph G, and further
- *perfect*, if each vertex $v \in V$ is *covered* by the considered matching M, i.e., there is a matching edge $\{v_1, v_2\} \in M$ with $v \in \{v_1, v_2\}$.

Reconsidering the example given in Figure 4.1, the subset of bold edges of the indistinguishability graph constitutes a matching. Although this matching is obviously maximum, it is *not* perfect as neither of the vertices $(\exists X)\, R(b, X, e)$ and $(\exists X)\, R(a, e, X)$ is an element of a matching edge. Considering only the connected component of this graph, whose vertices correspond to ground atoms of the given confidentiality policy, this connected component induces a subgraph on which the given maximum matching is even perfect.

4.1.2 Extending Maximum Clusterings

This insight that a perfect matching covering each of the cleaned policy elements can *not* be found on each indistinguishability graph immediately complies with the insight gained in Section 3.1.1 that – dependent on the set of admissible clusters

induced by a considered notion of admissible indistinguishabilities – it is *not* always possible to find a valid (non-extended) clustering covering each policy element and consisting only of non-singleton clusters. Instead, it may be necessary to artificially introduce additional potential secrets providing a basis for the construction of an extended clustering of only non-singleton clusters.

Correspondingly, an obvious idea is to create an extended clustering on the basis of a computed maximum matching M by simply pairing each potential secret uncovered by the matching M with a newly constructed additional potential secret. Again considering that a maximum matching already pairs as many elements of the considered (cleaned) confidentiality policy as possible, the number of additional potential secrets – artificially introducing knowledge to be additionally distorted – is minimized and hence availability is correspondingly maximized.

This pairing of uncovered policy elements with additional potential secrets is referred to as a *matching extension*, which is defined as follows.

Definition 4.3: Matching Extension

Let $\widehat{psec}$ be a cleaned confidentiality policy and suppose that M is a maximum matching on the indistinguishability graph corresponding to the set of admissible clusters over $\widehat{psec}$, which is induced by a given notion of admissible indistinguishabilities.

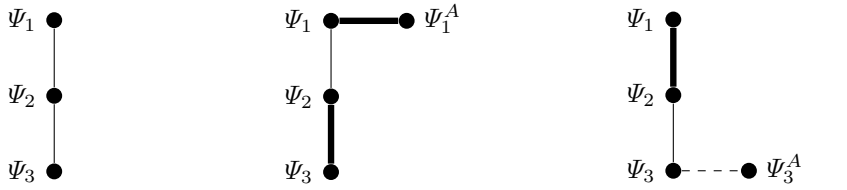
A *matching extension* M^* of the maximum matching M and the policy $\widehat{psec}$ under the given notion of admissible indistinguishabilities consists of

(i) each matching edge of M, i.e., $M \subseteq M^*$, and

(ii) a cluster $\{\Psi, \Psi^A\} \in M^*$ for each $\Psi \in \widehat{psec}$ not covered by M such that

- Ψ^A is an additional potential secret not occurring in $\widehat{psec}$ and
- M^* is an extended clustering of $\widehat{psec}$ with clusters of size 2 according to Definition 3.3, which obeys the given notion of admissible indistinguishabilities.

Although such a matching extension M^* is always an extended clustering, it restricts the construction of an extended clustering in the sense that this clustering must contain all clusters selected by the particular maximum matching – chosen from the set of possibly several maximum matchings on the corresponding indistinguishability graph – on which this matching extension is based. Hence, such a matching extension can only be constructed, if an additional potential secret Ψ^A



(a) Admissible clusters (b) Only extended clustering (c) Computed matching

Figure 4.2: Too weak well-defined notion of admissible indistinguishabilities

can be constructed for each potential secret Ψ uncovered by the given maximum matching such that the resulting cluster $\{\Psi, \Psi^A\}$ is admissible according to the given notion of admissible indistinguishabilities.

But even if a considered notion of admissible indistinguishabilities is supposed to be well-defined for clusters of size 2 in the sense of Definition 3.6, this requirement may be violated as exemplified in Figure 4.2. Under the supposition that the set of admissible clusters corresponding to the indistinguishability graph of Figure 4.2(a) is induced by this considered notion of admissible indistinguishabilities, Definition 3.6 guarantees that there is at least one extended clustering with clusters of size 2. Now, assume that solely the existence of the extended clustering given in Figure 4.2(b) – pairing the policy element Ψ_1 with the additional potential secret Ψ_1^A – is guaranteed and that the employed matching algorithm returns the maximum matching depicted in Figure 4.2(c) – pairing the policy element Ψ_1 with the other policy element Ψ_2. Then, the matching extension would require that the remaining policy element Ψ_3 is paired with an additional potential secret Ψ_3^A. But according to the assumption that solely the extended clustering of Figure 4.2(b) complies with the given notion of admissible indistinguishabilities, the resulting cluster $\{\Psi_3, \Psi_3^A\}$ would *not* be admissible.

To nonetheless always guarantee the existence of a matching extension independent of the computed maximum matching, the definition of well-defined notions of admissible indistinguishabilities can be strengthened as follows.

Definition 4.4: Well-Defined Indistinguishability (Strengthened)

Let $\widehat{psec}$ be the cleaned set of a given confidentiality policy *psec* and let G be the indistinguishability graph corresponding to the set of admissible clusters over $\widehat{psec}$, which is induced by a notion of admissible indistinguishabilities.

This notion of admissible indistinguishabilities is *well-defined* with respect to *psec*, if there is a matching extension M^* of *each* maximum matching M, which can be constructed on the indistinguishability graph G, such that

(i) there is a deterministic algorithm creating this matching extension M^* with all of its additional potential secrets Ψ^A with $\Psi^A \notin \widehat{psec}$,

(ii) the active domain of M^* is a finite subset of the domain Dom and

(iii) the domain Dom contains at least one constant symbol, which is *not* contained in the active domain of the matching extension M^*.

Similar to the algorithmic construction of the set of all admissible clusters over a given (cleaned) confidentiality policy (cf. Section 3.1.3), the development of an algorithm constructing an extended clustering M^* on the operational level – together with all of its additional potential secrets – crucially depends on the considered notion of admissible indistinguishabilities. Such an algorithm must hence be developed in accordance with this specific notion of admissible indistinguishabilities set up for the considered application scenario.

To guarantee that a constructed matching extension M^* satisfies all requirements of an extended clustering, particular attention must be paid that for each additional potential secret Ψ^A of this matching extension

- both non-implications $\Psi^A \not\models_{DB} \bar{\Psi}$ and $\bar{\Psi} \not\models_{DB} \Psi^A$ hold for each (possibly additional) potential secret $\bar{\Psi}$ occurring in a cluster of M^* and
- the corresponding additional cluster $\{\Psi, \Psi^A\}$ of M^* is admissible according to the considered notion of admissible indistinguishabilities.

Another challenge is that the construction of one additional cluster might limit the constructibility of other additional clusters. For example, suppose that the potential secrets Ψ_1 and Ψ_2 are *not* covered by a maximum matching and hence need to be paired with additional potential secrets. Moreover, suppose that

- Ψ_1 can be paired with the additional potential secrets Ψ_1^A and Ψ_2^A and
- Ψ_2 can solely be paired with the additional potential secret Ψ_2^A.

If the algorithm constructing the corresponding matching extension first chooses to construct the additional cluster $\{\Psi_1, \Psi_2^A\}$, the opportunity to afterwards also suitably pair Ψ_2 with an additional potential secret is gone.

At first sight, this example seems to suggest to again model the problem of constructing admissible additional clusters as a maximum matching problem on a (now even bipartite) graph, whose edges pair each potential secret Ψ uncovered

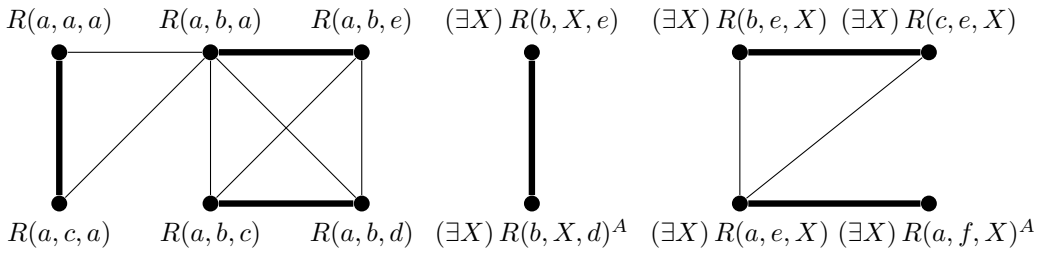


Figure 4.3: Indistinguishability graph with (bold) matching extension

by a maximum matching on the corresponding indistinguishability graph with each additional potential secret Ψ^A, which would allow for the construction of an admissible additional cluster $\{\Psi, \Psi^A\}$. But in general this approach is *not* feasible, as the selection of one additional potential secret for the construction of an additional cluster might exclude other additional potential secrets from being selectable for the construction of other additional clusters because of implication relationships between these additional potential secrets. Further considering that the employed domain *Dom* of constant symbols is supposed to be infinite, the number of additional potential secrets, with which an uncovered policy element can be paired, might also be infinite.

Reconsidering the maximum but *not* perfect matching M given in Figure 4.1, a matching extension M^* of M can be constructed as shown in Figure 4.3. This matching extension contains each edge of M and additionally pairs each of the policy elements $(\exists X)\, R(b,X,e)$ and $(\exists X)\, R(a,e,X)$, which are *not* covered by M, with an additional potential secret such that the resulting additional clusters comply with the interchangeability criterion (cf. Section 4.2.1) employed as a concrete notion of admissible indistinguishabilities. Considering the set

$$\widehat{psec}^* := \bigcup_{\{\Psi_1,\Psi_2\} \in M^*} \{\Psi_1, \Psi_2\}$$

of all (possibly additional) potential secrets of the matching extension M^* and the indistinguishability graph corresponding to the set of admissible clusters over $\widehat{psec}^*$ induced by the employed notion of admissible indistinguishabilities, the matching extension M^* is a perfect matching on this indistinguishability graph.

4.1.3 The Availability-Maximizing Weakening Algorithm

Now that all basic operations needed to actually compute an inference-proof weakened view – except for those whose algorithmic instantiation depends on the em-

ployed notion of admissible indistinguishabilities – are specified on the operational level, an instantiation of the generic algorithm introduced in Chapter 3 in the form of an availability-maximizing weakening algorithm can be presented. As already argued in Section 3.1.3, the employed notion of admissible indistinguishabilities must be adapted to the needs of each specific application scenario.

Algorithm 4.1: Inference-Proof Weakening (Availability-Max.)

Let r be a complete database instance over a database schema $\langle R|\mathcal{A}_R|\emptyset\rangle$ and let *psec* be a confidentiality policy of existentially quantified atoms. Moreover, suppose that a notion of admissible indistinguishabilities inducing admissible clusters of size 2 is given, which complies with the strengthened Definition 4.4 of well-defined notions of admissible indistinguishabilities.

Then, a weakened view $weak(r, psec)$ on r is created as follows:

- ***Stage 1*** *(independent of r): Disjoint clustering of potential secrets*
  - (i) Construct the cleaned set $\widehat{psec}$ based on *psec* (Def. 3.4)
  - (ii) Generate the indistinguishability graph $G = (V, E)$ (Def. 4.1)
  - (iii) Compute a maximum matching M on G (Def. 4.2)
  - (iv) Create the matching extension M^* of M and $\widehat{psec}$ (Def. 4.3)
- ***Stage 2*** *(dependent on r): Creation of weakened view*
  - (v) Create the subset $\mathcal{C}_r^* := \{\, C \in M^* \mid \mathcal{I}_r \models_M \bigvee_{\Psi \in C} \Psi \,\}$ of (extended) clusters containing a potential secret satisfied by $\mathcal{I}_r$
  - (vi) Create the weakened view $weak(r, psec)$ on r (Def. 3.7)

By applying Theorem 3.1, this algorithm returns an inference-proof weakened view in the sense that for each potential secret $\Psi \in psec$ the existence of at least one complete alternative instance r^Ψ is guaranteed, which

- obeys the considered potential secret Ψ, i.e., $\mathcal{I}_{r^\Psi} \not\models_M \Psi$, and
- guarantees that the corresponding weakened view $weak(r^\Psi, psec)$ is indistinguishable from $weak(r, psec)$, i.e., $weak(r^\Psi, psec) = weak(r, psec)$.

This theorem is applicable, as the availability-maximizing Algorithm 4.1 is equal to the generic Algorithm 3.1 except for the more concrete instantiation of the computation of an extended clustering within Stage 1. Further, each notion of

$$r = \{\, (a,b,c),\ (a,f,g),\ (b,a,e),\ (b,b,d),\ (b,d,f),\ (g,e,i),\ (g,h,i) \,\}$$

(a) Original database instance r

$$\begin{aligned} psec = \{\ & R(a,a,a),\ R(a,b,a),\ R(a,b,c),\ R(a,b,d),\ R(a,b,e),\ R(a,c,a), \\ & (\exists X)\, R(a,e,X),\ (\exists X)\, R(b,e,X),\ (\exists X)\, R(c,e,X),\ (\exists X)\, R(b,X,e)\ \} \end{aligned}$$

(b) Confidentiality policy $psec$ (already cleaned, i.e., $\widehat{psec} := psec$)

$\{\, R(a,b,c),\ R(a,b,d) \,\}$,
$\{\, (\exists X)\, R(b,X,e),\ (\exists X)\, R(b,X,d)^A \,\}$,
$\{\, (\exists X)\, R(a,e,X),\ (\exists X)\, R(a,f,X)^A \,\}$

(c) Set $\mathcal{C}_r^*$ of extended clusters of Figure 4.3, whose disjunctions are satisfied by $\mathcal{I}_r$

$\{\, R(a,a,a),\ R(a,c,a) \,\}$,
$\{\, R(a,b,a),\ R(a,b,e) \,\}$,
$\{\, (\exists X)\, R(b,e,X),\ (\exists X)\, R(c,e,X) \,\}$

(d) Set $M^* \setminus \mathcal{C}_r^*$ of extended clusters of Figure 4.3 (*not* satisfied by $\mathcal{I}_r$)

$R(b,d,f)$
$R(g,e,i)$
$R(g,h,i)$

$R(a,b,c) \vee R(a,b,d)$
$(\exists X)\, R(a,e,X) \vee (\exists X)\, R(a,f,X)$
$(\exists X)\, R(b,X,d) \vee (\exists X)\, R(b,X,e)$

$\neg\, [\, R(a,a,a) \vee R(a,c,a) \,]$
$\neg\, [\, R(a,b,a) \vee R(a,b,e) \,]$
$\neg\, [\, (\exists X)\, R(b,e,X) \vee (\exists X)\, R(c,e,X) \,]$

$$\begin{aligned} &(\forall X)(\forall Y)(\forall Z)\ [\\ &(\ X \equiv a \ \wedge\ Y \equiv b \ \wedge\ Z \equiv c\ ) \vee \\ &(\ X \equiv a \ \wedge\ Y \equiv b \ \wedge\ Z \equiv d\ ) \vee \\ &(\ X \equiv a \ \wedge\ Y \equiv e\ ) \vee \\ &(\ X \equiv a \ \wedge\ Y \equiv f\ ) \vee \\ &(\ X \equiv b \ \wedge\ Z \equiv d\ ) \vee \\ &(\ X \equiv b \ \wedge\ Z \equiv e\ ) \vee \\ &(\ X \equiv b \ \wedge\ Y \equiv d \ \wedge\ Z \equiv f\ ) \vee \\ &(\ X \equiv g \ \wedge\ Y \equiv e \ \wedge\ Z \equiv i\ ) \vee \\ &(\ X \equiv g \ \wedge\ Y \equiv h \ \wedge\ Z \equiv i\ ) \vee \\ &\neg R(X,Y,Z)\ ] \end{aligned}$$

(e) Weakened view $weak(r, psec)$ on r returned by Algorithm 4.1

Figure 4.4: Inference-proof weakening obeying the cleaned policy of Figure 4.1

admissible indistinguishabilities complying with the strengthened Definition 4.4 for well-defined notions of admissible indistinguishabilities – which essentially requires the existence of an extended clustering (in the form of a matching extension) containing all clusters induced by an arbitrary maximum matching – also complies with the non-strengthened Definition 3.6 – which essentially just requires the existence of at least one arbitrary extended clustering.

To give an example of the overall (availability-maximizing) weakening algorithm, consider the original database instance r given in Figure 4.4(a) and the (already cleaned) confidentiality policy $psec$ given in Figure 4.4(b). Moreover considering a notion of admissible indistinguishabilities called "interchangeability", which is introduced below in Section 4.2 and which provably complies with the strengthened

definition of well-defined notions of admissible indistinguishabilities, the indistinguishability graph given in Figure 4.1 corresponds to the set of induced admissible clusters over $\widehat{psec}$. The subset of bold edges of this graph is a maximum matching M, for which a matching extension M^* – corresponding to an extended clustering – can be determined as exemplified in Figure 4.3.

Within Stage 2 of the algorithm this determined extended clustering M^* is then partitioned into the subset $\mathcal{C}_r^*$ of those extended clusters of M^* given in Figure 4.4(c), whose corresponding disjunctions are satisfied by the considered original database instance r, and into the remaining subset $M^* \setminus \mathcal{C}_r^*$ of those extended clusters of M^* given in Figure 4.4(d), whose corresponding disjunctions are *not* satisfied by this instance r.

Now turning to the constructed weakened view $weak(r, psec)$ on r given in Figure 4.4(e), the positive knowledge $weak(r, psec)^+$ of this view only consists of those 3 tuples of the original instance r (represented as a lexicographically ordered sequence of corresponding ground atoms) *not* inducing a DB-Interpretation satisfying a disjunction, which corresponds to a cluster of $\mathcal{C}_r^*$. For each of these clusters $\{\Psi_1, \Psi_2\} \in \mathcal{C}_r^*$ the disjunctive knowledge $weak(r, psec)^\vee$ then presents the disjunction $\Psi_1 \vee \Psi_2$. Thereby, the sequence of the disjuncts Ψ_1 and Ψ_2 may be changed, dependent on the natural lexicographic order on these disjuncts, and the sequence on all of these disjunctions is also ordered lexicographically.

The presentation of the negative knowledge $weak(r, psec)^-$ then starts with each negated disjunction $\neg[\,\Psi_1 \vee \Psi_2\,]$ corresponding to a cluster $\{\Psi_1, \Psi_2\} \in M^* \setminus \mathcal{C}_r^*$ to ensure that an adversary's knowledge that neither of the sentences of a cluster of $M^* \setminus \mathcal{C}_r^*$ is satisfied by the original instance r is captured properly within the constructed weakened view on r (cf. Section 3.2). Thereby, these negated disjunctions are sorted just as the disjunctions of $weak(r, psec)^\vee$. The negative knowledge $weak(r, psec)^-$ is then concluded with the completeness sentence essentially expressing that each constant combination $(c_1, c_2, c_3) \in Dom^3$ *neither* inducing a DB-Interpretation satisfying a sentence of the positive knowledge *nor* inducing a DB-Interpretation satisfying any disjunct of the disjunctive knowledge is supposed to be *not* valid, i.e., $\neg R(c_1, c_2, c_3)$ is supposed to hold. Thereby, the arrangement of (and also within) the disjuncts of this completeness sentence is again inspired by the ordering of the disjunctive knowledge.

4.2 Admissible Indistinguishabilities by Local Distortion

So far, an availability-maximizing instantiation of the generic weakening algorithm has been developed. This instantiation specifies each of its basic operations, whose

algorithmic instantiation does *not* depend on the employed notion of admissible indistinguishabilities, on the operational level. For the construction of admissible clusters and the computation of a matching extension, whose algorithmic instantiation needs to be tailored to the employed notion of admissible indistinguishabilities, only coarse design guidelines could instead be given.

In fact, notions of admissible indistinguishabilities have only been discussed in a purely abstract way, yet. This discussion culminates in definitions declaratively describing requirements a possible notion of admissible indistinguishabilities should meet to be well-defined, but is far from giving any concrete implementable notion of admissible indistinguishabilities. On the one hand it is reasonable to keep possible notions of admissible indistinguishabilities as generic as possible and to thereby keep the weakening algorithm applicable for different application scenarios (cf. Section 3.1.3). On the other hand prototype implementations needed to evaluate the constructed weakening approach experimentally require a notion of admissible indistinguishabilities, which is concrete enough to be implemented.

4.2.1 Introducing Interchangeability

An example of an easy to implement notion of admissible indistinguishabilities for the construction of admissible clusters of size 2, which moreover complies with the strengthened definition of well-defined notions of admissible indistinguishabilities, is the so-called notion of *interchangeability*. This notion restricts distortion within a disjunction only locally in the sense that a pair of (possibly additional) potential secrets is *interchangeable*, if they differ in exactly one constant parameter. Note that the disjunctions of all examples given for the weakening approach so far are created in accordance with interchangeability.

Definition 4.5: Interchangeability

Two existentially quantified atoms $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ are *interchangeable*, if

(i) there is a *single differing position* $m \in \{1, \ldots, n\}$ such that both t_m and $\bar{t}_m$ are constant symbols of *Dom* with $t_m \neq \bar{t}_m$ and

(ii) for each other position $i \in \{1, \ldots, n\} \setminus \{m\}$ either

- both t_i and $\bar{t}_i$ are existentially quantified variables or
- both t_i and $\bar{t}_i$ are constant symbols of *Dom* with $t_i = \bar{t}_i$.

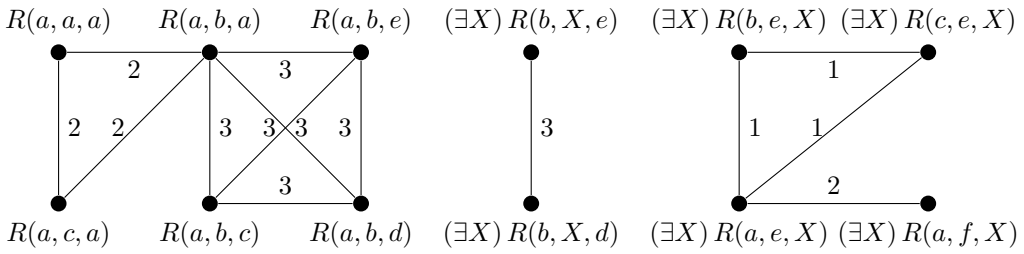


Figure 4.5: Indistinguishability graph complying with interchangeability

An example of a set of admissible clusters induced by this notion of interchangeability – represented in the form of an indistinguishability graph already known from Figure 4.3 – is given in Figure 4.5. For convenience, each edge of this graph is labeled with the single differing position of its incident potential secrets.

Note that all indistinguishability graphs resulting from interchangeability have the structure known from a specific class of graphs introduced by Knuth in [66]. The vertices of these graphs are supposed to be words of fixed length and a pair of these vertices is to be neighbored, if their corresponding words differ in exactly one character position. As further analyzed by Stiege in [83, 85], the vertices of each maximal clique of such a graph correspond to a maximal subset of words, which all pairwise differ in exactly one character at the same character position. Hence, each maximal clique of an indistinguishability graph resulting from interchangeability immediately induces a maximal subset of pairwise interchangeable potential secrets all sharing the same single differing position.

Inspired by this insight, the definition of interchangeability – which follows the spirit of this chapter and aims at constructing availability-maximizing clusters of size 2 – can naturally be extended to be also applicable for the construction of larger admissible clusters of size $k \geq 2$. Then, a subset of existentially quantified atoms is interchangeable, if all of its sentences are pairwise interchangeable and all of these pairs moreover share the same single differing position m. This leads to clusters inducing disjunctions of the form

$$\bigvee_{i \in \{1,\ldots,k\}} (\exists \boldsymbol{X})\, R(t_1, \ldots, t_{m-1}, \tilde{c}_m^{(i)}, t_{m+1}, \ldots, t_n)$$

with each term t_j occurring in each of the disjuncts and being either a constant symbol of *Dom* or a variable of $\boldsymbol{X}$ and with each term $\tilde{c}_m^{(i)}$ occurring solely in the i-th disjunct and being a constant symbol of *Dom* such that $\tilde{c}_m^{(i_1)} \neq \tilde{c}_m^{(i_2)}$ holds for each pair $1 \leq i_1 < i_2 \leq k$ of disjuncts. Such a disjunction restricts distortion only locally within this disjunction by revealing the knowledge that the original

database instance satisfies the sentence template

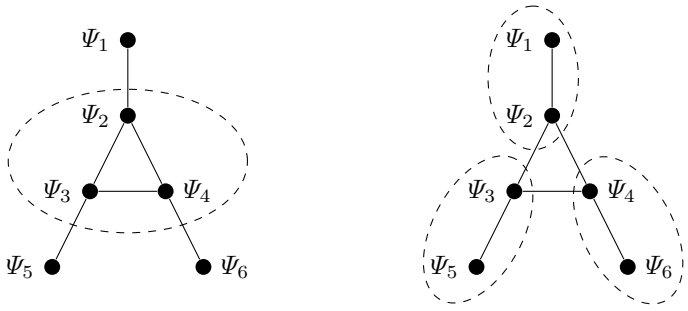
$$(\exists \boldsymbol{X})\, R(t_1, \ldots, t_{m-1}, \square, t_{m+1}, \ldots, t_n)$$

and only hides with which (non-empty subset) of the constant values $\tilde{c}_m^{(1)}, \ldots, \tilde{c}_m^{(k)}$ the m-th term $\square$ of this sentence template is actually instantiated.

This generalization of a specific value to a wider set of possible values is known from the well-known approaches of k-anonymization and ℓ-diversification [47, 70, 79, 86], which motivated the usage of weakening disjunctions – instead of less cooperative refusals – to distort confidential knowledge (cf. Section 1.3.1). Hence, this connection between the weakening algorithm developed in this thesis instantiated with the interchangeability criterion and the approaches of k-anonymization and ℓ-diversification might allow the construction of k-anonymized and ℓ-diverse instances with the help of a suitably adapted version of the weakening algorithm. This is of particular interest when considering also further a priori knowledge an adversary might have (as done in Chapter 5), as current standard approaches to k-anonymization and ℓ-diversification do *not* handle an adversary's a priori knowledge in a formal way.

On the downside, a disadvantage of this notion of interchangeability clearly is that it only provides a suitable number of admissible disjunction templates, if a given confidentiality policy contains a lot of potential secrets structurally *not* differing much from each other. If this is *not* the case, a large number of additional potential secrets – each of which requires additional knowledge to be distorted – may be needed to construct a suitable disjunction template for each of the (cleaned) policy elements and hence employing this notion of admissible indistinguishabilities may even result in a loss of availability. This is also experimentally confirmed by Experiment 2 of Section 6.3 and demonstrates that the task of suitably defining a concrete notion of admissible indistinguishabilities crucially depends on the specific application scenario considered.

Reconsidering that an indistinguishability graph resulting from interchangeability can be decomposed into a set of maximal cliques covering this graph and that each of these maximal cliques consists of a maximal subset of pairwise interchangeable potential secrets all sharing the same single differing position, this property seems to induce a straightforward algorithm for the construction of clusters of a size larger than 2. Additionally considering that this decomposition into maximal cliques is also computationally efficient on graphs of this special structure [84, 85], this approach seems to be even more promising. But as (maximal) cliques of such a graph do generally *not* need to be pairwise vertex-disjoint [85] (cf. Figure 4.5) – while it is mandatory that (extended) clusters inducing disjunction templates are pairwise disjoint (cf. Section 3.2) – it is still to decide to which cluster each of the



(a) Non-optimal solution (b) Optimal solution

Figure 4.6: Constructing clusters of size 3 on graphs based on interchangeability

vertices occurring in more than one clique should be assigned, thereby minimizing the number of additional potential secrets needed to extend too small clusters.

This problem is exemplified in Figure 4.6. As depicted in Figure 4.6(a), the given indistinguishability graph contains the (maximal) clique $\{\Psi_2, \Psi_3, \Psi_4\}$ and following the above mentioned ideas this clique might induce a corresponding cluster of size 3. Consequently, each of the remaining policy elements Ψ_1, Ψ_5 and Ψ_6 needs to be paired with 2 additional potential secrets to obtain an extended clustering with clusters of size 3. Instead choosing the three disjoint cliques depicted in Figure 4.6(b), each of the (too-small) induced clusters of size 2 needs to be complemented with only one additional potential secret to obtain clusters of size 3, thereby resulting in an extended clustering with less additional potential secrets.

On a graph, on which the construction of an extended clustering with clusters of size 3 is possible without introducing any additional potential secrets, the problem of constructing an extended clustering with clusters of size 3 obviously corresponds to the decision problem of whether a given graph can be partitioned into a certain given number of (vertex-)disjoint triangles, as (sub-)cliques of size 3 correspond to triangles. This problem is well-known to be NP-complete for general graphs [58] and seems to be NP-complete for the above mentioned subclass of graphs resulting from the interchangeability property, too. But nonetheless, these insights might provide a promising basis for the construction of reasonable heuristic solutions.

4.2.2 Well-Defined Interchangeability

Now returning to the construction of availability-maximizing disjunctions consisting of 2 disjuncts, it is still to show that the interchangeability criterion complies

with the strengthened Definition 4.4 for well-defined notions of admissible indistinguishabilities. In fact, the even stronger property that interchangeability is well-defined according to Definition 4.4 for *each* possible confidentiality policy is shown, as long as this policy does *not* contain a sentence semantically equivalent to the weakest possible potential secret $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$ without any constant symbols. This specific notion of admissible indistinguishabilities is hence applicable independent of the structure such a policy has.

As already elaborated in Section 3.1.2, the construction of an extended clustering – and hence also the construction of a matching extension being an extended clustering – is in general only possible for confidentiality policies *not* containing a potential secret semantically equivalent to the weakest possible existentially quantified atom $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$. As a consequence, there can *not* be any notion of admissible indistinguishabilities, which is well-defined for confidentiality policies containing such a potential secret, as the well-definedness of such a notion essentially requires the constructibility of an extended clustering.

Now, the well-definedness of interchangeability is formally captured in the following theorem. Thereby, the proof of this theorem is constructive in the sense that it sketches an efficient as well as easy to implement algorithm for the construction of matching extensions under interchangeability.

Theorem 4.1: Well-Defined Interchangeability

Let $\widehat{psec}$ be the cleaned set of an arbitrary confidentiality policy *psec*, which does *not* contain a sentence semantically equivalent to the weakest possible potential secret $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$ without any constant symbols. Further, consider the indistinguishability graph G corresponding to the set of admissible clusters over $\widehat{psec}$ induced by interchangeability as known from Definition 4.5.

This notion of interchangeability implements a *well-defined* notion of admissible indistinguishabilities with respect to *psec* in the sense that the existence of a matching extension M^* of *each* maximum matching M, which can be constructed on G, is guaranteed such that

(i) there is a deterministic algorithm creating this matching extension M^* with all of its additional potential secrets Ψ^A with $\Psi^A \notin \widehat{psec}$,

(ii) the active domain of M^* is a finite subset of the domain *Dom* and

(iii) the domain *Dom* contains at least one constant symbol, which is *not* contained in the active domain of the matching extension M^*.

Proof. To prove the existence of a matching extension constructed by a deterministic algorithm, consider the cleaned set $\widehat{psec}$ of a given (finite) confidentiality policy $psec$ and a finite set $\widehat{psec}^A$ of (possibly already constructed) additional potential secrets such that the union $\widehat{psec} \cup \widehat{psec}^A$ is a cleaned set. Then, a (further) additional potential secret $\Psi^A = (\exists \boldsymbol{X})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ obeying the interchangeability criterion can be constructed deterministically as follows for each potential secret $\Psi = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ of the cleaned policy $\widehat{psec}$:

- choose a differing position $m \in \{1, \ldots, n\}$ with $t_m \in Dom$ deterministically,
- set $\bar{t}_j := t_j$ for each $j \in \{1, \ldots, n\} \setminus \{m\}$ and
- deterministically choose $\bar{t}_m$ to be a constant of Dom such that $\bar{t}_m \neq \tilde{t}_m$ holds for each $(\exists \boldsymbol{Y})\, R(\tilde{t}_1, \ldots, \tilde{t}_n) \in \widehat{psec} \cup \widehat{psec}^A$ with $\tilde{t}_m \in Dom$.

Note that a differing position $m \in \{1, \ldots, n\}$ with $t_m \in Dom$ can always be found as $psec$ (and hence also $\widehat{psec}$) does *not* contain a sentence semantically equivalent to the weakest possible potential secret $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$ without any constant symbols. Additionally considering that the domain Dom of constant symbols is supposed to be *infinite* and that the union $\widehat{psec} \cup \widehat{psec}^A$ is supposed to be a *finite* set of existentially quantified atoms, a "fresh" constant symbol as required above can always be found for each admissible differing position m with $t_m \in Dom$.

Next, consider a maximum matching M constructed on the indistinguishability graph corresponding to the set of admissible clusters over $\widehat{psec}$, which are induced by interchangeability. Then, assuming that the elements of $\widehat{psec}$ are ordered in a deterministic way, a matching extension M^* of M and $\widehat{psec}$ under interchangeability can always be constructed deterministically as follows:

- initially, set $\widehat{psec}^A := \emptyset$ and $M^* := M$,
- then, for each policy element $\Psi_i \in \widehat{psec}$ uncovered by M, one after another according to their assumed order,
  - construct an additional potential secret Ψ_i^A as described above and
  - add the cluster $\{\Psi_i, \Psi_i^A\}$ to M^* and Ψ_i^A to $\widehat{psec}^A$.

This construction, which leads to a finite matching extension M^* and which further relies on the above mentioned construction of additional potential secrets using only constant symbols occurring in policy elements of $\widehat{psec}$ (and hence stemming from Dom) or "fresh" constant symbols of Dom *not* occurring in $\widehat{psec}$, obviously guarantees that the active domain of M^* is a finite subset of Dom. As a direct consequence, the *infinite* domain Dom contains at least one constant symbol, which is *not* contained in the *finite* active domain of the matching extension M^*.

Reconsidering that a matching extension must be an extended clustering according to Definition 3.3, the constructed M^* obviously is such an extended clustering, provided that $\widehat{psec} \cup \widehat{psec}^A$ is cleaned as required by Definition 3.5. To prove that $\widehat{psec} \cup \widehat{psec}^A$ is cleaned after each construction step of M^*, consider an arbitrary potential secret $\Psi_i = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n) \in \widehat{psec}$ uncovered by the maximum matching M and an arbitrary set $\widehat{psec}^A$ of (possibly already constructed) additional potential secrets such that the union $\widehat{psec} \cup \widehat{psec}^A$ is a cleaned set – which is in particular true for $\widehat{psec}^A = \emptyset$ because of $\widehat{psec}$ being a cleaned set.

Now, suppose that in one construction step the above sketched algorithm constructs an interchangeable additional potential secret $\Psi_i^A = (\exists \boldsymbol{X})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ for the considered policy element Ψ_i such that the union $\widehat{psec} \cup \widehat{psec}^A \cup \{\Psi_i^A\}$ violates the cleaned set property, i.e., there is a (possibly additional) potential secret $\tilde{\Psi} = (\exists \boldsymbol{Z})\, R(\tilde{t}_1, \ldots, \tilde{t}_n) \in \widehat{psec} \cup \widehat{psec}^A$ with $\tilde{\Psi} \models_{DB} \Psi_i^A$ or $\Psi_i^A \models_{DB} \tilde{\Psi}$. Considering that the above given construction of Ψ_i^A requires the term $\bar{t}_m$ of Ψ_i^A to be a constant symbol of Dom with $\bar{t}_m \neq \tilde{t}_m$, the implication $\tilde{\Psi} \models_{DB} \Psi_i^A$ can *not* hold according to Lemma 2.1. Under the assumption that the other implication $\Psi_i^A \models_{DB} \tilde{\Psi}$ holds, Lemma 2.1 requires that for each term $\tilde{t}_j$ of $\tilde{\Psi}$, which is a constant symbol of Dom, the term $\bar{t}_j$ of Ψ_i^A is also a constant symbol of Dom with $\bar{t}_j = \tilde{t}_j$. Hence, the term $\tilde{t}_m$ of $\tilde{\Psi}$ must be a variable of $\boldsymbol{Z}$, as $\bar{t}_m \neq \tilde{t}_m$ is supposed to hold by construction of Ψ_i^A. For each other term $\tilde{t}_j$ of $\tilde{\Psi}$ with $\tilde{t}_j \in Dom$, for which $\tilde{t}_j = \bar{t}_j$ is supposed to hold, the equality $\tilde{t}_j = t_j$ must also hold for the term t_j of Ψ_i as $\bar{t}_j = t_j$ is supposed to hold by construction of Ψ_i^A. So, by again applying Lemma 2.1, the validity of the implication $\Psi \models_{DB} \tilde{\Psi}$ is a direct consequence in contradiction to the assumption that $\widehat{psec} \cup \widehat{psec}^A$ is a cleaned set. ♠

CHAPTER 5

Introducing A Priori Knowledge

The generic weakening algorithm and its availability-maximizing instantiation have been developed to enforce a confidentiality policy in the sense that an adversary is provably *not* able to infer that the knowledge embodied in a potential secret of this policy is satisfied by a considered original database instance – even if this adversary tries to recover confidential knowledge with the help of logical reasoning. Thereby, he might employ his knowledge about the released weakened view on the original database instance as well as his awareness of the algorithm used to construct this weakened view – parameterized with all inputs except for the original database instance to be protected – as a basis for his logical reasoning.

But until now, an adversary is *not* supposed to have any further a priori knowledge about semantic database constraints possibly arranged for the schema underlying the original database instance or about the world in general. This additional knowledge is of particular interest, as it might open up extended possibilities to draw confidentiality compromising conclusions. In the following, semantic database constraints stemming from a restricted subclass of well-known "Tuple Generating Dependencies" are considered and analyzed with respect to an adversary's extended possibilities to draw harmful inferences. Then, an adapted version of the availability-maximizing weakening algorithm, which provably eliminates these additional inference-channels, is provided.

5.1 A Subclass of Tuple Generating Dependencies

As mentioned above, an adversary might from now on have some a priori knowledge *prior* consisting of semantic database constraints stemming from a restricted subclass of Tuple Generating Dependencies (cf. [1, 54]). The constraints of this subclass, which are referred to as *single premise tuple generating dependencies*, are supposed to be arranged for the schema underlying a considered original database instance. As usual, each database instance has to satisfy all database constraints arranged for its underlying schema [1].

Definition 5.1: Single Premise Tuple Generating Dependency

A sentence Γ of the first-order language $\mathscr{L}$ is a *single premise tuple generating dependency* over a predicate symbol R of arity n, if it is of the form

$$(\forall \boldsymbol{X})\,[\,R(t_1,\ldots,t_n) \Rightarrow (\exists \boldsymbol{Y})\,R(\bar{t}_1,\ldots,\bar{t}_n)\,]$$

and furthermore

(i) each term t_i of the premise of Γ is either a constant symbol of Dom or a universally quantified variable of $\boldsymbol{X}$, i.e., $t_i \in \boldsymbol{X} \cup Dom$,

(ii) each term $\bar{t}_i$ of the conclusion of Γ is either a constant symbol of Dom, a universally quantified variable of $\boldsymbol{X}$ or an existentially quantified variable of $\boldsymbol{Y}$, i.e., $\bar{t}_i \in \boldsymbol{X} \cup \boldsymbol{Y} \cup Dom$,

(iii) the set $\boldsymbol{X}$ of variables is $\boldsymbol{X} = \{t_1,\ldots,t_n\} \setminus Dom$,

(iv) the set $\boldsymbol{Y}$ of variables is $\boldsymbol{Y} = \{\bar{t}_1,\ldots,\bar{t}_n\} \setminus (Dom \cup \boldsymbol{X})$,

(v) each variable of $\boldsymbol{X}$ can occur at most once in $R(t_1,\ldots,t_n)$, i.e., $t_i \neq t_j$ for all $t_i, t_j \in \{t_1,\ldots,t_n\} \setminus Dom$ with $i \neq j$, and

(vi) each variable of $\boldsymbol{X} \cup \boldsymbol{Y}$ can occur at most once in $R(\bar{t}_1,\ldots,\bar{t}_n)$, i.e., $\bar{t}_i \neq \bar{t}_j$ for all $\bar{t}_i, \bar{t}_j \in \{\bar{t}_1,\ldots,\bar{t}_n\} \setminus Dom$ with $i \neq j$.

As later discussed in Section 5.3, it might be worthwhile to additionally require that at least one term t_i of the premise of Γ and at least one term $\bar{t}_j$ of the conclusion of Γ is a constant symbol of Dom.

The semantics of single premise tuple generating dependencies is the same as for (the more general) tuple generating dependencies, which are well-known from the area of relational databases: if the premise of such a dependency Γ is satisfied

under a constant substitution σ – substituting the universally quantified variables of $\boldsymbol{X}$ with constant symbols of Dom – by a DB-Interpretation $\mathcal{I}$, the conclusion of this dependency Γ must be satisfied under this constant substitution σ by the DB-Interpretation $\mathcal{I}$, too.

A single premise tuple generating dependency hence is a tuple generating dependency, which is syntactically restricted such that its premise must *not* contain more than one (conjunctively connected) atom and that – considering a constant substitution of the universally quantified variables of $\boldsymbol{X}$ – both the premise and the conclusion of such a dependency are existentially quantified atoms in the sense of Definition 2.3. Hence, the validity of DB-Implications, in which a premise or a conclusion of such a dependency is involved, can be decided efficiently as known from Lemma 2.1 by reducing the (generally hard to solve) implication problem under DB-Semantics to an efficiently decidable pattern matching problem.

To be able to handle single premise tuple generating dependencies more conveniently in the remainder of this chapter, the following conventions are used.

Definition 5.2: Handling of Single Premise TGDs

Suppose that Γ is a single premise tuple generating dependency of the form $(\forall \boldsymbol{X})\,[\,R(t_1,\ldots,t_n) \Rightarrow (\exists \boldsymbol{Y})\,R(\bar{t}_1,\ldots,\bar{t}_n)\,]$. Then,

(i) $prem\,(\Gamma) := (\exists \boldsymbol{X})\,R(t_1,\ldots,t_n)$ is the *existentially quantified premise* of the dependency Γ and

(ii) $concl\,(\Gamma) := (\exists \boldsymbol{Y})(\exists \boldsymbol{Z})\,R(\bar{t}_1,\ldots,\bar{t}_n)$ with $\boldsymbol{Z} := \boldsymbol{X} \cap \{\bar{t}_1,\ldots,\bar{t}_n\}$ is the *existentially quantified conclusion* of the dependency Γ.

Moreover, consider σ to be a *constant substitution* of the variables of $\boldsymbol{X}$ in the form of a function $\sigma : \boldsymbol{X} \cup \boldsymbol{Y} \cup Dom \to \boldsymbol{Y} \cup Dom$ with

$$\sigma(\tilde{t}_i) := \begin{cases} u_X \in Dom, & \text{if } \tilde{t}_i = X \text{ and } X \in \boldsymbol{X} \\ \tilde{t}_i, & \text{if } \tilde{t}_i \in \boldsymbol{Y} \text{ or } \tilde{t}_i \in Dom \end{cases}$$

substituting each variable $X \in \boldsymbol{X}$ occurring in $prem\,(\Gamma)$ and in $concl\,(\Gamma)$ with a constant symbol $u_X \in Dom$, i.e., $\sigma(X) = u_X$. Then,

(i) $prem\,(\Gamma)[\sigma] := R(\sigma(t_1),\ldots,\sigma(t_n))$ is the *premise* of the dependency Γ under the constant substitution σ and

(ii) $concl\,(\Gamma)[\sigma] := (\exists \boldsymbol{Y})\,R(\sigma(\bar{t}_1),\ldots,\sigma(\bar{t}_n))$ is the *conclusion* of the dependency Γ under the constant substitution σ.

For the sake of convenience, a constant substitution σ – which actually aims at substituting the (originally universally quantified) variables of $\boldsymbol{X}$ with constant symbols of Dom – is also applicable for constant symbols of Dom and variables of $\boldsymbol{Y}$ by simply mapping each $u \in Dom$ and each $Y \in \boldsymbol{Y}$ to itself, i.e., $\sigma(u) = u$ and $\sigma(Y) = Y$. This guarantees that σ is applicable for each term $\tilde{t}_i$ of a premise and a conclusion of a single premise tuple generating dependency without distincting which type of term $\tilde{t}_i$ actually is.

Similarly to the observation that DB-Implication can be decided efficiently for existentially quantified atoms (cf. Lemma 2.1), it is also easy to decide whether there is a constant substitution under which a given existentially quantified atom implies another existentially quantified atom under DB-Semantics. As shown in the following, this (extended) implication problem can again be reduced to an efficiently decidable pattern matching problem.

Lemma 5.1: DB-Implication under Constant Substitution

Suppose that the existentially quantified atoms $(\exists \boldsymbol{X})(\exists \boldsymbol{Y})\, R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Z})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ are given. There is a constant substitution σ substituting the variables of $\boldsymbol{X}$ with constant symbols of Dom (and mapping variables of $\boldsymbol{Y}$ and constants of Dom to themselves) such that the DB-Implication

$$(\exists \boldsymbol{Y})\, R(\sigma(t_1), \ldots, \sigma(t_n)) \models_{DB} (\exists \boldsymbol{Z})\, R(\bar{t}_1, \ldots, \bar{t}_n)$$

holds, *if and only if* for each term $\bar{t}_i$, which is a constant symbol of Dom, the (unsubstituted) term t_i is either a variable of $\boldsymbol{X}$ or a constant symbol of Dom such that $t_i = \bar{t}_i$.

Proof. To start with the *only-if-part*, consider the given existentially quantified atoms $(\exists \boldsymbol{X})(\exists \boldsymbol{Y})\, R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Z})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ and suppose that for each term $\bar{t}_i$, which is a constant symbol of Dom, the (unsubstituted) term t_i is either a variable of $\boldsymbol{X}$ or a constant symbol of Dom such that $t_i = \bar{t}_i$. Moreover, consider the constant substitution $\sigma : \boldsymbol{X} \cup \boldsymbol{Y} \cup Dom \to \boldsymbol{Y} \cup Dom$ with

$$\sigma(t_i) := \begin{cases} \bar{t}_i, & \text{if } t_i \in \boldsymbol{X} \text{ and } \bar{t}_i \in Dom \\ u_i \in Dom, & \text{if } t_i \in \boldsymbol{X} \text{ and } \bar{t}_i \in \boldsymbol{Z} \\ t_i, & \text{if } t_i \in \boldsymbol{Y} \text{ or } t_i \in Dom \end{cases}$$

substituting each variable $X \in \boldsymbol{X}$ of the sentence $(\exists \boldsymbol{X})(\exists \boldsymbol{Y})\, R(t_1, \ldots, t_n)$ with a constant symbol of Dom. As a direct consequence of the construction of σ, for each term $\bar{t}_i$, which is a constant symbol of Dom, the (substituted) term $\sigma(t_i)$

now is a constant symbol of *Dom* such that $\sigma(t_i) = \bar{t}_i$. Hence, the DB-Implication $(\exists \boldsymbol{Y})\, R(\sigma(t_1), \ldots, \sigma(t_n)) \models_{DB} (\exists \boldsymbol{Z})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ holds under the constructed constant substitution σ according to Lemma 2.1.

To now prove the *if-part* by contraposition, again consider the given existentially quantified atoms $(\exists \boldsymbol{X})(\exists \boldsymbol{Y})\, R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Z})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ and suppose that there is an $m \in \{1, \ldots, n\}$ such that $\bar{t}_m$ is a constant symbol of *Dom* and the (unsubstituted) term t_m is *neither* a variable of $\boldsymbol{X}$ *nor* a constant symbol of *Dom* with $t_m = \bar{t}_m$. Hence, t_m is either a variable of $\boldsymbol{Y}$ or a constant symbol of *Dom* with $t_m \neq \bar{t}_m$. As the constant substitution σ is supposed to map variables of $\boldsymbol{Y}$ and constants of *Dom* to themselves, $\sigma(t_m) = t_m$ is supposed to hold under *each* possible constant substitution σ. Thus, considering the sentences

$$(\exists \boldsymbol{Y})\, R(\sigma(t_1), \ldots, \sigma(t_m), \ldots, \sigma(t_n)) \text{ and } (\exists \boldsymbol{Z})\, R(\bar{t}_1, \ldots, \bar{t}_m, \ldots, \bar{t}_n)$$

under an *arbitrary* constant substitution σ, the term $\bar{t}_m$ is a constant symbol of *Dom* and the (substituted) term $\sigma(t_m)$ is either a variable of $\boldsymbol{Y}$ or a constant symbol of *Dom* with $\sigma(t_m) \neq \bar{t}_m$. Therefore, by again applying Lemma 2.1, the DB-Implication $(\exists \boldsymbol{Y})\, R(\sigma(t_1), \ldots, \sigma(t_n)) \models_{DB} (\exists \boldsymbol{Z})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ does *not* hold under *each* possible constant substitution σ. ♠

5.2 Adapting the Clustering of Confidentiality Policies

Now that an adversary's a priori knowledge *prior* is supposed to contain some knowledge in the form of single premise tuple generating dependencies, it is to analyze if and to what extent an adversary might employ this additional knowledge to compromise a confidentiality policy with the help of logical reasoning. This analysis should lead to the development of counter measures provably disabling these harmful additional reasoning capabilities by suitably restricting the knowledge, which a weakened view reveals to an adversary.

5.2.1 Confidentiality Compromising Interferences

As each dependency declared for a database schema imposes restrictions on the structure of database instances over this schema, an adversary might exploit his knowledge about these restrictions to exclude some alternative database instances *not* obeying these restrictions from being the original database instance of his interest, which is publicly known to obey these restrictions – although these alternative instances seem (from the adversary's point of view) to be indistinguishable from a considered original database instance when neglecting the knowledge about

these restrictions valid database instances must obey. If all remaining credible (alternative) database instances then satisfy a certain piece of knowledge, whose validity is to be kept secret according to a confidentiality policy, the adversary finally succeeds in compromising this confidentiality policy (cf. [7]).

As a first example of harmful a priori knowledge, suppose that *prior* contains a single premise tuple generating dependency

$$\Gamma \;=\; (\forall X)\,[\,R(X,b,c) \Rightarrow R(d,e,X)\,]\;,$$

whose existentially quantified premise $\mathit{prem}\,(\Gamma) = (\exists X)\,R(X,b,c)$ is implied by a potential secret Ψ_1, i.e., $\Psi_1 \models_{DB} \mathit{prem}\,(\Gamma)$, of a confidentiality policy *psec* with

$$\mathit{psec} \;=\; \{\,\Psi_1 = R(a,b,c),\; \Psi_2 = R(a,b,d)\,\}\;.$$

Further, suppose that the released weakened view contains the sequences

- $\mathit{weak}\,(r, \mathit{psec})^{+} = \emptyset$ and
- $\mathit{weak}\,(r, \mathit{psec})^{\vee} = \{\,R(a,b,c) \vee R(a,b,d)\,\}$.

If the potential secret $\Psi_2 = R(a,b,d)$ is chosen to be protected, an alternative instance r^{Ψ_2} constructed to obey Ψ_2 must *not* contain the tuple (a,b,d). Otherwise, $\mathcal{I}_{r^{\Psi_2}} \models_M \Psi_2$ would hold and the alternative instance r^{Ψ_2} would *not* obey Ψ_2. To further guarantee $\mathcal{I}_{r^{\Psi_2}} \models_M \Psi_1 \vee \Psi_2$, the alternative instance r^{Ψ_2} must contain the tuple (a,b,c) to satisfy Ψ_1. Moreover, other tuples different from (a,b,c) and (a,b,d) can *not* be in r^{Ψ_2} as otherwise the property of indistinguishability would be violated due to $\mathit{weak}\,(r, \mathit{psec})^{+} = \emptyset \neq \mathit{weak}\,(r^{\Psi_2}, \mathit{psec})^{+}$. Hence, neglecting the adversary's a priori knowledge *prior* for now, $r^{\Psi_2} = \{\,(a,b,c)\,\}$ is the only possibility to construct an alternative instance obeying Ψ_2 from the adversary's point of view. But taking his a priori knowledge *prior* into account, the adversary can exclude r^{Ψ_2} from being the "real" original instance of his interest: the validity of $\mathcal{I}_{r^{\Psi_2}} \models_M \Gamma$ does *not* hold because of $\mathcal{I}_{r^{\Psi_2}} \models_M \mathit{prem}\,(\Gamma)[\sigma]$ under a constant substitution σ with $\sigma(X) = a$ and because of further $\mathcal{I}_{r^{\Psi_2}} \not\models_M \mathit{concl}\,(\Gamma)[\sigma]$ under this constant substitution. Hence, the adversary's knowledge is sufficient to conclude that the original instance r of his interest must contain the tuple (a,b,d) and to thereby violate the confidentiality policy.

To give a second example of harmful a priori knowledge, suppose that *prior* again contains the single premise tuple generating dependency

$$\Gamma \;=\; (\forall X)\,[\,R(X,b,c) \Rightarrow R(d,e,X)\,]\;,$$

for which the implication $\mathit{concl}(\Gamma)[\sigma] \models_{DB} \Psi_1$ now holds under a constant substitution σ with $\sigma(X) = a$ for a potential secret Ψ_1 stemming from the policy

$$\mathit{psec} \;=\; \{\, \Psi_1 = (\exists X)\, R(d, X, a),\; \Psi_2 = (\exists X)\, R(d, X, b) \,\} \;\; .$$

Further, suppose that the released weakened view contains the sequences

- $\mathit{weak}(r, \mathit{psec})^{+} = \{\, R(a, b, c) \,\}$ and
- $\mathit{weak}(r, \mathit{psec})^{\vee} = \{\, (\exists X)\, R(d, X, a) \vee (\exists X)\, R(d, X, b) \,\}$.

If the potential secret $\Psi_1 = (\exists X)\, R(d, X, a)$ is chosen to be protected, the alternative instance r^{Ψ_1} constructed to obey Ψ_1 must *not* contain any tuple $\boldsymbol{c} \in \mathit{Dom}^n$ which induces a DB-Interpretation $\mathcal{I}_{\boldsymbol{c}}$ with $\mathcal{I}_{\boldsymbol{c}}(R) = \{\boldsymbol{c}\}$ satisfying Ψ_1. As the sentence $R(a, b, c)$ is in $\mathit{weak}(r, \mathit{psec})^{+}$, this sentence $R(a, b, c)$ must also be in $\mathit{weak}(r^{\Psi_1}, \mathit{psec})^{+}$ to preserve indistinguishability from the adversary's point of view. The corresponding database tuple (a, b, c) must hence be in *each* alternative instance r^{Ψ_1} and as a consequence $\mathcal{I}_{r^{\Psi_1}}$ satisfies $\mathit{prem}(\Gamma)[\sigma]$ under a constant substitution σ with $\sigma(X) = a$. So, $\mathcal{I}_{r^{\Psi_1}} \models_M \mathit{concl}(\Gamma)[\sigma]$ must hold, too, to guarantee $\mathcal{I}_{r^{\Psi_1}} \models_M \Gamma$. But adding the tuple $\mathit{concl}(\Gamma)[\sigma] = (d, e, a)$ to r^{Ψ_1} immediately results in $\mathcal{I}_{r^{\Psi_1}} \models_M \Psi_1$. Consequently, the adversary can conclude that each database instance satisfying both $\mathit{weak}(r, \mathit{psec})$ and *prior* – and hence also the original instance r of his interest – must contain the tuple (d, e, a), thereby violating the potential secret Ψ_1.

More generally, each of the two example setups above enables an adversary to infer sensitive knowledge, as the considered dependency of *prior interferes* with a potential secret of a disjunction $\Psi_1 \vee \Psi_2$ due to a corresponding implication relationship. In general, the purpose of this disjunction $\Psi_1 \vee \Psi_2$ is to weaken sensitive knowledge of the original instance to such an extent that a larger set of (alternative) databases instances with different truth values for both Ψ_1 and Ψ_2 becomes credible from an adversary's point of view. But due to the interference with the dependency of *prior*, the adversary is nonetheless able to infer the "real" truth values of some elements the considered confidentiality policy by excluding a specific subset of alternative instances from being credible, which is actually needed to protect these compromised policy elements.

In a scenario *without* a priori knowledge, the inference-proofness of each weakened view $\mathit{weak}(r, \mathit{psec})$ is essentially achieved by strictly isolating the disjunctive knowledge of $\mathit{weak}(r, \mathit{psec})^{\vee}$ – which aims at *not* revealing the real truth values of the disjuncts of $\mathit{weak}(r, \mathit{psec})^{\vee}$ to an adversary – from the definite knowledge an adversary can gain about the original database instance (cf. Section 3.2). This isolation follows the goal that the satisfaction or non-satisfaction of a disjunct of $\mathit{weak}(r, \mathit{psec})^{\vee}$ can *not* be concluded based on the satisfaction or non-satisfaction

of a piece of definite knowledge the adversary is aware of. But as soon as a priori knowledge in the form of single premise tuple generating dependencies interfering with disjuncts of $weak(r, psec)^{\vee}$ is additionally considered, this isolation can be broken up in the sense that a dependency bridges the gap between a piece of definite knowledge and a certain disjunct and thereby allows for reasoning about the satisfaction or non-satisfaction of this disjunct.

Both examples discussed above deal with interference due to implication relationships. But the conclusion of a single premise tuple generating dependency can even harmfully interfere with a potential secret in scenarios without such an implication relationship. Actually, the existence of a so-called common constant unifier – inspired by the concept of unifiers known from first-order logic [76, 88] – may be sufficient for harmful interferences.

Definition 5.3: Common Constant Unifier

Consider two existentially quantified atoms Φ and Ψ both constructed over the same predicate symbol R of arity n. These sentences Φ and Ψ share a *common constant unifier* $\boldsymbol{c}$, if there is a constant combination $\boldsymbol{c} \in Dom^n$ such that the induced DB-Interpretation $\mathcal{I}_{\boldsymbol{c}}$ with $\mathcal{I}_{\boldsymbol{c}}(R) = \{\boldsymbol{c}\}$ satisfies both Φ and Ψ, i.e., $\mathcal{I}_{\boldsymbol{c}} \models_M \Phi$ and $\mathcal{I}_{\boldsymbol{c}} \models_M \Psi$.

Note that this definition of a common constant unifier, which is based on the satisfaction of sentences, slightly deviates from commonly found definitions of (most general) unifiers known from literature on first-order logic, which are based on the substitution of variables – such as in [76, 88], for example. But considering commonly used definitions of satisfaction of first-order logic, this difference vanishes considerably: a common constant unifier $\boldsymbol{c} = (c_1, \ldots, c_n)$ for two existentially quantified atoms $\Phi = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and $\Psi = (\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ induces a DB-Interpretation $\mathcal{I}_{\boldsymbol{c}}$ satisfying both Φ and Ψ. Taking moreover into account that $\mathcal{I}_{\boldsymbol{c}} \models_M \Phi$ (and $\mathcal{I}_{\boldsymbol{c}} \models_M \Psi$, respectively) only holds, if each variable t_i with $t_i \in \boldsymbol{X}$ can be substituted by the constant c_i of $\boldsymbol{c}$ (analogously for each $\bar{t}_i \in \boldsymbol{Y}$), a common constant unifier induces a possible substitution of variables in the sense of commonly found definitions of (most general) unifiers.

To now give an example that harmful interference is also possible, if the conclusion of a single premise tuple generating dependency just shares a common constant unifier with a potential secret of a confidentiality policy, consider the following example setup, which only slightly deviates from the second example setup in the given single premise tuple generating dependency:

- $\Gamma \;=\; (\forall X)\,[\,R(X,b,c) \Rightarrow (\exists Y)\,R(Y,e,X)\,]$ is a dependency of *prior*,
- $psec \;=\; \{\,\Psi_1 = (\exists X)\,R(d,X,a),\; \Psi_2 = (\exists X)\,R(d,X,b)\,\}$ is the policy, and
- the weakened view $weak(r, psec)$ contains the sequences
  - $weak(r, psec)^{+} = \{\,R(a,b,c)\,\}$ and
  - $weak(r, psec)^{\vee} = \{\,(\exists X)\,R(d,X,a) \vee (\exists X)\,R(d,X,b)\,\}$.

Obviously, there is *neither* an implication relationship between $prem(\Gamma)[\sigma]$ and a policy element of *psec* under an arbitrary constant substitution σ, *nor* an implication relationship between $concl(\Gamma)[\sigma]$ and a policy element of *psec* under an arbitrary constant substitution σ. Instead, $concl(\Gamma)$ and Ψ_1 just share the common constant unifier (d, e, a).

If the potential secret $\Psi_1 = (\exists X)\,R(d,X,a)$ is chosen to be protected, the alternative instance r^{Ψ_1} constructed to obey Ψ_1 must *not* contain the tuple (d, e, a), which induces a DB-Interpretation satisfying Ψ_1. As known from the second example, the tuple (a, b, c) must be in *each* alternative instance r^{Ψ_1} protecting Ψ_1 to preserve indistinguishability and as a consequence $\mathcal{I}_{r^{\Psi_1}}$ satisfies $prem(\Gamma)[\sigma]$ under a constant substitution σ with $\sigma(X) = a$. But adding an arbitrary tuple to r^{Ψ_1}, which induces a DB-Interpretation satisfying $concl(\Gamma)[\sigma]$, either results in adding the tuple (d, e, a) satisfying Ψ_1 or in adding a tuple $\boldsymbol{c} \in Dom^n$ with both $\mathcal{I}_{\boldsymbol{c}} \models_M concl(\Gamma)[\sigma]$ and $\mathcal{I}_{\boldsymbol{c}} \not\models_M \Psi_1$, which violates indistinguishability because of both $R(\boldsymbol{c}) \in weak(r^{\Psi_1}, psec)^{+}$ and $R(\boldsymbol{c}) \notin weak(r, psec)^{+}$. Consequently, the adversary can again conclude that each database instance satisfying both $weak(r, psec)$ and *prior* – and hence also the original instance r of his interest – must contain the tuple (d, e, a), thereby violating the potential secret Ψ_1.

To be able to efficiently decide on the existence of common constant unifiers on the operational level, the following insight is of importance. Similar to the implication problems considered in Lemma 2.1 and Lemma 5.1, the existence of a common constant unifier can be decided with the help of an efficiently decidable pattern matching problem.

Lemma 5.2: Existence of Common Constant Unifiers

Suppose that both $(\exists \boldsymbol{X})\,R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Y})\,R(\bar{t}_1, \ldots, \bar{t}_n)$ are existentially quantified atoms. These sentences share a common constant unifier, *if and only if* the equality $t_i = \bar{t}_i$ holds for each $i \in \{1, \ldots, n\}$ with both $t_i \in Dom$ and $\bar{t}_i \in Dom$.

Proof. To start with the *only-if-part*, consider the given existentially quantified atoms $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ and suppose that the equality $t_i = \bar{t}_i$ holds for each $i \in \{1, \ldots, n\}$ with both $t_i \in \mathit{Dom}$ and $\bar{t}_i \in \mathit{Dom}$. Next, consider the constant combination $\boldsymbol{c} = (c_1, \ldots, c_n)$ with

- $c_i = t_i = \bar{t}_i$ for each $i \in \{1, \ldots, n\}$ with both $t_i \in \mathit{Dom}$ and $\bar{t}_i \in \mathit{Dom}$,
- $c_i = t_i$ for each $i \in \{1, \ldots, n\}$ with $t_i \in \mathit{Dom}$ and $\bar{t}_i \in \boldsymbol{Y}$,
- $c_i = \bar{t}_i$ for each $i \in \{1, \ldots, n\}$ with $t_i \in \boldsymbol{X}$ and $\bar{t}_i \in \mathit{Dom}$, and
- $c_i \in \mathit{Dom}$ (arbitrarily) for each $i \in \{1, \ldots, n\}$ with $t_i \in \boldsymbol{X}$ and $\bar{t}_i \in \boldsymbol{Y}$.

This constant combination $\boldsymbol{c}$ can always be constructed as $t_i = \bar{t}_i$ is supposed to hold for each $i \in \{1, \ldots, n\}$ with both $t_i \in \mathit{Dom}$ and $\bar{t}_i \in \mathit{Dom}$ and the induced DB-Interpretation $\mathcal{I}_{\boldsymbol{c}}$ with $\mathcal{I}_{\boldsymbol{c}}(R) = \{\boldsymbol{c}\}$ obviously satisfies both $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$.

To now prove the *if-part* by contraposition, again consider the given existentially quantified atoms $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ and assume that there is an index $m \in \{1, \ldots, n\}$ with

$$t_m \in \mathit{Dom} \;\text{ and }\; \bar{t}_m \in \mathit{Dom} \quad \text{and with further} \quad t_m \neq \bar{t}_m \;\; .$$

The construction of a DB-Interpretation $\mathcal{I}_{\boldsymbol{c}}$ with $\mathcal{I}_{\boldsymbol{c}}(R) = \{\boldsymbol{c}\}$ satisfying both of the existentially quantified atoms $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and $(\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ can only be succeeded, if there is constant combination $\boldsymbol{c} = (c_1, \ldots, c_n)$ with both $c_m = t_m$ and $c_m = \bar{t}_m$. But such a constant combination $\boldsymbol{c}$ can *not* be found as $t_m \neq \bar{t}_m$ is supposed to hold. ♠

Reconsidering the above given examples of how confidentiality requirements can be breached on the basis of single premise tuple generating dependencies interfering with confidentiality policies, a formal definition of *interference* between an adversary's a priori knowledge and a confidentiality policy is now provided. While the conditions (i) and (iii) of this definition are clearly motivated by the examples presented above, the reason for condition (ii) will become clear when later proving a property of a so-called partitioning of an adversary's a priori knowledge, which is captured in Lemma 5.3 and needed for the final proof of inference-proofness of an extended version of the weakening algorithm.

Moreover, note that the definition of DB-Implication guarantees that existentially quantified atoms always share a common constant unifier, if there is an implication relationship between them. Hence, the above mentioned case of interference due to an implication relationship between the conclusion of a dependency and a potential secret is covered by condition (iii) of the following definition.

Definition 5.4: Interference

Let *psec* be a confidentiality policy and let *prior* be an adversary's a priori knowledge consisting of single premise tuple generating dependencies. A dependency $\Gamma \in \mathit{prior}$ *interferes* with the policy *psec*, if there is a potential secret $\Psi \in \mathit{psec}$ such that

(i) the implication $\Psi \models_{DB} \mathit{prem}(\Gamma)$ holds,

(ii) the implication $\Psi[\bar{\sigma}] \models_{DB} \mathit{prem}(\Gamma)$ holds under an arbitrary constant substitution $\bar{\sigma}$, provided that there is a dependency $\bar{\Gamma} \in \mathit{prior}$ with $\Psi = \mathit{concl}(\bar{\Gamma})$, or

(iii) Ψ and $\mathit{concl}(\Gamma)$ share a common constant unifier.

An adversary's a priori knowledge *prior* is said to *interfere* with a confidentiality policy *psec*, if there is a dependency $\Gamma \in \mathit{prior}$ interfering with *psec*.

Note that interference between a dependency and a confidentiality policy does *not* necessarily mean that this confidentiality policy can actually be compromised. To exemplify this, suppose that Ψ_1, Ψ_2, Ψ_3 and Ψ_4 are ground atoms over the same predicate symbol R and that

- $\mathit{prior} = \{ \Psi_1 \Rightarrow \Psi_2 \}$ is an adversary's a priori knowledge and
- $\mathit{psec} = \{ \Psi_1, \Psi_2, \Psi_3, \Psi_4 \}$ is a given confidentiality policy.

For now, further suppose that the clustering algorithm pairs the potential secrets Ψ_1 and Ψ_2 to one cluster and the potential secrets Ψ_3 and Ψ_4 to another cluster and that the released weakened view $\mathit{weak}(r, \mathit{psec})$ contains the sequences

- $\mathit{weak}(r, \mathit{psec})^{+} = \emptyset$ and
- $\mathit{weak}(r, \mathit{psec})^{\vee} = \{ \Psi_1 \vee \Psi_2,\ \Psi_3 \vee \Psi_4 \}$ and
- the negative knowledge $\mathit{weak}(r, \mathit{psec})^{-}$ reveals that all knowledge different from Ψ_1, Ψ_2, Ψ_3 and Ψ_4 is *not* satisfied by the original instance r.

If the potential secret Ψ_2 is chosen to be protected, each alternative instance r^{Ψ_2} obeying Ψ_2 must *not* satisfy Ψ_2 and must hence satisfy Ψ_1 to satisfy the disjunction $\Psi_1 \vee \Psi_2$. As a consequence of this construction, the dependency $\Psi_1 \Rightarrow \Psi_2$ of the adversary's a priori knowledge *prior* is *not* satisfied by $\mathcal{I}_{r^{\Psi_2}}$. This allows the adversary to exclude each alternative instance obeying Ψ_2 from being real and to thereby compromise the confidentiality policy by concluding that Ψ_2 must be satisfied by the considered original instance.

Now assuming that the original instance r underlying the example does *neither* satisfy Ψ_1 nor Ψ_2, the released weakened view $weak(r, psec)$ contains the sequences

- $weak(r, psec)^+ = \emptyset$ and
- $weak(r, psec)^\vee = \{ \Psi_3 \vee \Psi_4 \}$ and
- the negative knowledge $weak(r, psec)^-$ reveals that all knowledge different from Ψ_3 and Ψ_4 is *not* satisfied by the original instance r.

Under this modified setup alternative database instances, which protect an arbitrary element of the given confidentiality policy *psec* and which are moreover consistent with both $weak(r, psec)$ and *prior*, can easily be constructed: such an instance must satisfy one of the potential secrets Ψ_3 and Ψ_4 (in accordance with the potential secret to be protected) and must *not* satisfy any further knowledge.

If the clustering stage of the algorithm instead chooses to pair the potential secrets Ψ_1 and Ψ_3 to one cluster and the potential secrets Ψ_2 and Ψ_4 to another cluster, this different clustering might lead to a weakened view $weak(r, psec)$ containing the sequences

- $weak(r, psec)^+ = \emptyset$ and
- $weak(r, psec)^\vee = \{ \Psi_1 \vee \Psi_3,\ \Psi_2 \vee \Psi_4 \}$ and
- the negative knowledge $weak(r, psec)^-$ reveals that all knowledge different from Ψ_1, Ψ_2, Ψ_3 and Ψ_4 is *not* satisfied by the original instance r.

Under this setup alternative database instances achieving consistency with both $weak(r, psec)$ and *prior* can again easily be constructed:

- to protect Ψ_1 such an instance must satisfy Ψ_3 and at least one of the disjuncts Ψ_2 and Ψ_4 and must *not* satisfy any further knowledge;
- to protect Ψ_2 such an instance must satisfy Ψ_4 and Ψ_3 and must *not* satisfy any further knowledge;
- to protect Ψ_3 such an instance must satisfy Ψ_1 and Ψ_2 and must *not* satisfy any further knowledge except for Ψ_4;
- to protect Ψ_4 such an instance must satisfy Ψ_2 and at least one of the disjuncts Ψ_1 and Ψ_3 and must *not* satisfy any further knowledge.

Hence, the given definition of interference is *not* precise enough to decide whether confidential knowledge can *actually* be inferred by an adversary exploiting his a priori knowledge. An interference between a given dependency and a given confidentiality policy instead only indicates that such an inference-channel *might* possibly exist under certain (extended) clusterings of the confidentiality policy

and under certain original database instances. The given definition of interference can hence *not* serve as a basis for an algorithm distorting only actually harmful knowledge – thereby achieving best availability – but might still provide a basis for the construction of reasonable heuristic solutions.

5.2.2 Extending the Confidentiality Policy

Considering a single premise tuple generating dependency Γ of an adversary's a priori knowledge *prior*, which interferes with a confidentiality policy *psec*, the next obvious question is how to limit this adversary's knowledge such that the dependency Γ does *not* possibly enable him to compromise the confidentiality policy. A first obvious idea is to extend the confidentiality policy *psec* by the potential secret $\Psi_{\text{prem}} := \mathit{prem}(\Gamma)$ to be able to weaken knowledge, which satisfies the premise of Γ and thereby imposes the additional requirement that the conclusion of Γ must also be satisfied under the constant substitution(s) satisfying $\mathit{prem}(\Gamma)$.

As this weakening of the introduced potential secret Ψ_{prem} should be enforced with the help of a disjunction $\Psi_{\text{prem}} \vee \Psi$ pairing Ψ_{prem} with another (possibly additional) potential secret Ψ, the construction of an alternative instance r^{Ψ} obeying this other potential secret Ψ must generally be possible. But such an alternative instance r^{Ψ} has to satisfy Ψ_{prem} instead of Ψ to satisfy the disjunction $\Psi_{\text{prem}} \vee \Psi$. Thus, as a consequence of so-called *forward chaining*, this alternative instance must also satisfy the conclusion of Γ under the constant substitution(s) satisfying $\mathit{prem}(\Gamma)$ – even if the considered original instance does *neither* satisfy the premise *nor* the conclusion of Γ. To be always able to construct an alternative instance consistent with Γ without making this alternative instance (from an adversary's point of view) distinguishable from the original instance, the confidentiality policy *psec* should also be extended by the potential secret $\Psi_{\text{concl}} = \mathit{concl}(\Gamma)$ to be able to add database tuples to an alternative instance, which do *not* stem from the original instance but induce DB-Interpretations satisfying the conclusion of Γ.

Another idea to mitigate the impact of Γ is to extend the confidentiality policy *psec* by *only* adding the potential secret $\Psi_{\text{concl}} := \mathit{concl}(\Gamma)$ to be able to consistently add database tuples to satisfy the conclusion of Γ. But if this potential secret Ψ_{concl} is to be obeyed by an alternative instance, this instance must *not* satisfy the conclusion of Γ. Thus, as a consequence of so-called *backward chaining*, the premise of Γ must *not* be satisfied by this instance, either – even if it is satisfied by the original instance. To be again able to construct an alternative instance consistent with Γ, the confidentiality policy *psec* should also be extended by the potential secret $\Psi_{\text{prem}} = \mathit{prem}(\Gamma)$ to be also able to suitably weaken the knowledge about a possible satisfaction of Ψ_{prem}.

Summing up these insights, it is generally *not* sufficient to extend a confidentiality policy *psec* by only one of the potential secrets $prem(\Gamma)$ and $concl(\Gamma)$ to prevent an adversary from drawing harmful inferences by exploiting a dependency Γ of his a priori knowledge, which interferes with the policy *psec*. Instead, the confidentiality policy *psec* should be extended by both of these potential secrets $prem(\Gamma)$ and $concl(\Gamma)$.

Considering an adversary's a priori knowledge containing *multiple* single premise tuple generating dependencies, such an extension of a confidentiality policy can of course lead to new interferences between the dependencies of this a priori knowledge and the extended part of the confidentiality policy. For instance, suppose that the adversary's a priori knowledge *prior* contains the dependencies

- $\Gamma_1 \;=\; (\forall X)\,[\,R(X,a,b) \Rightarrow R(a,b,X)\,]$ and
- $\Gamma_2 \;=\; (\forall X)\,[\,R(a,X,c) \Rightarrow R(a,X,d)\,]$

and further suppose that the (non-extended) policy is $psec = \{\, R(a,a,b)\,\}$. Obviously, the dependency Γ_1 interferes with *psec* because of $R(a,a,b) \models_{DB} prem(\Gamma_1)$ and the dependency Γ_2 does *not* interfere with *psec* because of the non-implication $R(a,a,b) \not\models_{DB} prem(\Gamma_2)$ and the absence of a common constant unifier between $R(a,a,b)$ and $concl(\Gamma_2)$. The extended confidentiality policy $psec_{prior}$ then still contains the potential secret $R(a,a,b)$ stemming from the (non-extended) original confidentiality policy *psec* and additionally contains the potential secrets $prem(\Gamma_1)$ and $concl(\Gamma_1)$, i.e.,

$$psec_{prior} \;=\; \{\, R(a,a,b),\; (\exists X)\,R(X,a,b),\; (\exists X)\,R(a,b,X)\,\}\;.$$

Now, the dependency Γ_2 interferes with this extended policy $psec_{prior}$ because $concl(\Gamma_2)$ and the potential secret $(\exists X)\,R(a,b,X)$ of $psec_{prior}$ share the common constant unifier (a,b,d). As a consequence, the already extended confidentiality policy $psec_{prior}$ must again be extended by adding the potential secrets $prem(\Gamma_2)$ and $concl(\Gamma_2)$ to $psec_{prior}$, i.e.,

$$\begin{aligned} psec_{prior} = \{\; & R(a,a,b), \\ & (\exists X)\,R(X,a,b),\; (\exists X)\,R(a,b,X), \\ & (\exists X)\,R(a,X,c),\; (\exists X)\,R(a,X,d)\;\}\;. \end{aligned}$$

As a consequence of this insight that extensions of a confidentiality policy can lead to new interferences, an algorithm extending a confidentiality policy must hence iteratively extend this possibly already (partly) extended policy with respect to the given a priori knowledge until a fixpoint is reached and hence *no* further extension of this policy is possible.

This is captured in the following operational definition of an extended confidentiality policy, which employs the temporary variable $psec'_{prior}$ to check within each iteration of the policy extension, whether a fixpoint is reached.

Definition 5.5: Extended Confidentiality Policy

Let *psec* be a confidentiality policy and let *prior* be an adversary's a priori knowledge consisting of single premise tuple generating dependencies.

The *extended* confidentiality policy $psec_{prior}$, which is an *extension* of *psec* with respect to *prior*, is constructed by

(i) initially setting $psec_{prior} := psec$ and $psec'_{prior} := \emptyset$ and

(ii) then, as long as $psec_{prior} \neq psec'_{prior}$ holds, by repeatedly computing

(a) $psec'_{prior} := psec_{prior}$ and

(b) $psec_{prior} := psec_{prior} \cup \{\, prem(\Gamma),\ concl(\Gamma) \,\}$ for each dependency Γ stemming from *prior* and interfering with $psec'_{prior}$.

After a confidentiality policy is extended with respect to an adversary's a priori knowledge, it can be cleaned as known from Section 3.1.2. Reconsidering the extended policy $psec_{prior}$ given above, the corresponding extended and cleaned confidentiality policy is

$$\widehat{psec}_{prior} = \{\ (\exists X)\, R(X,a,b),\ (\exists X)\, R(a,b,X),\\ (\exists X)\, R(a,X,c),\ (\exists X)\, R(a,X,d)\ \}\ .$$

Although the (only) original policy element $R(a,a,b)$ of *psec* is *not* contained in the extended and cleaned policy $\widehat{psec}_{prior}$ any more, it is still implicitly protected by the weaker element $(\exists X)\, R(X,a,b) \in \widehat{psec}_{prior}$ as known from Lemma 3.1.

5.2.3 Enforcing the Satisfaction of Dependencies

Now that an extended and cleaned confidentiality policy $\widehat{psec}_{prior}$ contains both a potential secret Ψ_{prem} with $prem(\Gamma) \models_{DB} \Psi_{\text{prem}}$ and a potential secret Ψ_{concl} with $concl(\Gamma) \models_{DB} \Psi_{\text{concl}}$ for each dependency Γ of an adversary's a priori knowledge *prior* interfering with the extended policy $psec_{prior}$, it is to discuss how the knowledge revealed to an adversary can be weakened in such a way that this adversary is *not* able to exploit his a priori knowledge harmfully.

Considering solely one single dependency $\Gamma \in \mathit{prior}$ for now, knowledge satisfying the potential secret Ψ_{prem} can in general be suitably weakened by replacing it with a disjunction $\Psi_{\text{prem}} \vee \Psi$ pairing Ψ_{prem} with another (possibly additional) potential secret Ψ. This disjunctive knowledge does *not* imply the potential secret Ψ_{prem} and does hence – as a consequence of Lemma 3.1 – *not* imply the (stronger) sentence $\mathit{prem}(\Gamma)$, either. But if $\mathit{prem}(\Gamma)$ and Ψ_{prem} are semantically equivalent, additional care must be taken to ensure that this other potential secret Ψ is *not* semantically equivalent to Ψ_{concl}.[4] Otherwise, the knowledge $\Psi_{\text{prem}} \vee \Psi_{\text{concl}}$ is introduced and there is hence *no* alternative instance obeying Ψ_{concl}: such an instance *not* satisfying Ψ_{concl} must instead satisfy Ψ_{prem} (and hence also $\mathit{prem}(\Gamma)$) to satisfy $\Psi_{\text{prem}} \vee \Psi_{\text{concl}}$ and thus the satisfaction of the premise of Γ immediately leads to the requirement that the conclusion of Γ must also be satisfied.

The potential secret $\mathit{concl}(\Gamma)$ has been added to the extended policy $\mathit{psec}_{\mathit{prior}}$ with the idea in mind that it should be possible to consistently add database tuples to an alternative instance, which make this alternative instance satisfy the conclusion of Γ (cf. Section 5.2.2). But according to the semantics of potential secrets, the potential secret $\mathit{concl}(\Gamma)$ should just prevent an adversary from getting to know that an original instance satisfies $\mathit{concl}(\Gamma)$. If an original instance does *not* satisfy $\mathit{concl}(\Gamma)$, this knowledge may instead be revealed to an adversary. Accordingly, the approach developed so far only constructs disjunctions for clusters having at least one potential secret satisfied by a considered original instance. For all other clusters, the non-satisfaction of their potential secrets is revealed, as this knowledge is *not* protected by the confidentiality policy and as the construction of corresponding disjunctions would even result in knowledge *not* consistent with the original instance. Hence, a possible non-satisfaction of the conclusion of Γ might still be revealed to an adversary.

As a first obvious solution to this problem one could try to add the negated sentence $\neg\mathit{concl}(\Gamma)$ instead of $\mathit{concl}(\Gamma)$ to the extended confidentiality policy. But the sentence $\neg\mathit{concl}(\Gamma)$ is *not* a feasible policy element, as confidentiality policies are supposed to consist of only (non-negated) existentially quantified atoms according to Definition 2.4 to be able to decide on the validity of DB-Implications between these existentially quantified atoms with the help of an efficiently decidable pattern matching problem (cf. Lemma 2.1).

Another idea is to prevent the construction of a *not* satisfied disjunction template containing the policy element Ψ_{concl} of $\widehat{\mathit{psec}}_{\mathit{prior}}$, independent of whether a considered original database instance actually satisfies a disjunct of this disjunction or

[4] Considering the below mentioned result that potential secrets semantically equivalent to Ψ_{concl} must *not* occur in regular disjunctions, this scenario can actually *not* occur, but is nonetheless presented to provide a comprehensive insight into the problem of weakening Ψ_{prem}.

not to keep the clustering stage of the weakening algorithm instance-independent (cf. Section 3.1.1). This requirement of instance-independent satisfaction – together with the requirement that weakened views should contain only true knowledge – immediately leads to the construction of a disjunction pairing Ψ_{concl} with tautological knowledge, i.e., a disjunction semantically equivalent to $\Psi_{\text{concl}} \vee \mathit{true}$. Such a disjunction leads to a maximum weakening (or generalization) of Ψ_{concl} and hence corresponds to the complete refusal of Ψ_{concl}, i.e., *no* knowledge about the satisfaction or non-satisfaction of the sentence Ψ_{concl} is revealed at all.

Then, neglecting other elements of the confidentiality policy $\widehat{psec}_{prior}$ for now, the sentence $\mathit{concl}(\Gamma)[\sigma]$ can be satisfied under *any* possible constant substitution σ. Due to the assumed validity of the implication $\mathit{concl}(\Gamma) \models_{DB} \Psi_{\text{concl}}$ the stronger sentence $\mathit{concl}(\Gamma)[\sigma]$ with $\mathit{concl}(\Gamma)[\sigma] \models_{DB} \mathit{concl}(\Gamma)$ also implies Ψ_{concl}, i.e., $\mathit{concl}(\Gamma)[\sigma] \models_{DB} \Psi_{\text{concl}}$. Hence, the definition of DB-Implication guarantees the existence of a database tuple, which induces a DB-Interpretation satisfying both $\mathit{concl}(\Gamma)[\sigma]$ and Ψ_{concl}. Such a database tuple can be added consistently to an alternative database instance, as all knowledge about possible non-existences of original database tuples satisfying Ψ_{concl} is refused.

But now considering that an extended and cleaned confidentiality policy $\widehat{psec}_{prior}$ usually contains *multiple* potential secrets, the consistency of an alternative instance may still get lost. For instance, suppose that $\widehat{psec}_{prior}$ also contains a potential secret Ψ with $\Psi \neq \Psi_{\text{concl}}$, for which the implication $\mathit{concl}(\Gamma)[\sigma] \models_{DB} \Psi$ is supposed to hold under a certain constant substitution σ. Moreover, suppose that Ψ is in a cluster containing only potential secrets *not* satisfied by a considered original instance. Hence, this non-satisfaction of Ψ is revealed to an adversary. If then $\mathit{concl}(\Gamma)[\sigma]$ needs to be satisfied by an alternative instance – due to $\mathit{prem}(\Gamma)[\sigma]$ being satisfied – the implication $\mathit{concl}(\Gamma)[\sigma] \models_{DB} \Psi$ would require that Ψ is satisfied by this alternative instance, too – in contradiction to the adversary's knowledge that Ψ is *not* satisfied by the original instance. To mitigate this inference-channel, Ψ needs to be refused just as Ψ_{concl}.

To exemplify that those potential secrets of $\widehat{psec}_{prior}$, which are only implied by a conclusion of a dependency under a certain constant substitution of this conclusion – and *not* under "normal" DB-Implication – actually need to be refused, consider the original database instance

$$r = \{\ (a, c, a)\ \}\ ,$$

an adversary's a priori knowledge *prior* consisting of the dependencies

- $\Gamma_1 \ = \ (\forall X)\,[\,R(a, b, X) \Rightarrow R(d, e, c)\,]$ and
- $\Gamma_2 \ = \ (\forall X)\,[\,R(d, e, X) \Rightarrow R(X, b, a)\,]$

and the (already) extended and cleaned confidentiality policy

$$\widehat{psec}_{prior} = \{\ (\exists X)\, R(a,c,X),\ (\exists X)\, R(a,b,X),\ (\exists X)\, R(d,e,X),\\ (\exists X)\, R(X,b,a),\ (\exists X)\, R(c,b,X),\ (\exists X)\, R(c,d,X)\ \}\ .$$

Now, assume that only those potential secrets of $\widehat{psec}_{prior}$, which are implied by an existentially quantified conclusion of one of the dependencies Γ_1 and Γ_2 in the sense of "normal" DB-Implication, are actually refused. Then,

- $(\exists X)\, R(d,e,X)$ is to be refused due to $concl(\Gamma_1) \models_{DB} (\exists X)\, R(d,e,X)$ and
- $(\exists X)\, R(X,b,a)$ is to be refused due to $concl(\Gamma_2) \models_{DB} (\exists X)\, R(X,b,a)$,

but in particular $(\exists X)\, R(c,b,X)$ does *not* need to be refused because of both

- $concl(\Gamma_1) \not\models_{DB} (\exists X)\, R(c,b,X)$ and
- $concl(\Gamma_2) \not\models_{DB} (\exists X)\, R(c,b,X)$.

This potential secret $(\exists X)\, R(c,b,X)$ is hence clustered and might be grouped together with $(\exists X)\, R(c,d,X)$. But as both of these potential secrets are *not* satisfied by the original instance r, both of these non-satisfactions are revealed. Further, suppose that the potential secret $(\exists X)\, R(a,c,X)$ is paired with $(\exists X)\, R(a,b,X)$ such that the knowledge about the satisfaction of $(\exists X)\, R(a,c,X)$ is weakened by the corresponding disjunction $(\exists X)\, R(a,b,X) \vee (\exists X)\, R(a,c,X)$.

Now, assume that the potential secret $\Psi = (\exists X)\, R(a,c,X)$ is chosen to be protected. As an alternative instance r^{Ψ} obeying this policy element must *not* satisfy Ψ, this alternative instance must instead satisfy $(\exists X)\, R(a,b,X)$ to nonetheless satisfy the disjunction $(\exists X)\, R(a,b,X) \vee (\exists X)\, R(a,c,X)$. Hence, r^{Ψ} must contain a tuple $(a,b,\square)$ with an arbitrary constant symbol $\square \in Dom$. To satisfy the dependency Γ_1, whose premise is satisfied by r^{Ψ} due to $(a,b,\square) \in r^{\Psi}$, this alternative instance r^{Ψ} must further contain the tuple (d,e,c) to also satisfy the conclusion of Γ_1. Analogously, the tuple (c,b,a) now needs to be added to r^{Ψ} to guarantee that r^{Ψ} satisfies Γ_2. As a direct consequence, the potential secret $(\exists X)\, R(c,b,X)$ is now also satisfied by r^{Ψ} in contradiction to the adversary's knowledge that the original instance r does *not* satisfy $(\exists X)\, R(c,b,X)$. This enables the adversary to exclude each alternative instance protecting $(\exists X)\, R(a,c,X)$ from being the "real" instance r of his interest.

To identify those potential secrets of an extended and cleaned confidentiality policy $\widehat{psec}_{prior}$, which need to be refused completely because of being implied by the conclusion of a dependency under an arbitrary constant substitution, the following subset $concl(\widehat{psec}_{prior}, prior)$ of $\widehat{psec}_{prior}$ is created.

Definition 5.6: Potential Secrets to be Refused Completely

Let *prior* be an adversary's a priori knowledge consisting of single premise tuple generating dependencies and let $\widehat{psec}_{prior}$ be a confidentiality policy, which has first been extended with respect to *prior* and then been cleaned.

Then, $concl(\widehat{psec}_{prior}, prior)$ is the subset

$$\{\, \Psi \in \widehat{psec}_{prior} \mid \text{there is a } \Gamma \in prior, \text{ for which } concl(\Gamma)[\sigma] \models_{DB} \Psi \text{ holds under an arbitrary constant substitution } \sigma \,\}$$

of potential secrets of $\widehat{psec}_{prior}$.

Considering an arbitrary dependency Γ of an adversary's a priori knowledge *prior* and the set $\widehat{psec}_{prior} \setminus concl(\widehat{psec}_{prior}, prior)$ of those potential secrets, which are *not* implied by the conclusion of a dependency of *prior* under an arbitrary constant substitution, it is possible to construct a database tuple, which induces a DB-Interpretation satisfying $concl(\Gamma)[\sigma]$ under an arbitrary constant substitution σ without satisfying *any* potential secret Ψ of $\widehat{psec}_{prior} \setminus concl(\widehat{psec}_{prior}, prior)$. Thereby, the existence of such a database tuple is essentially guaranteed by the non-implication $concl(\Gamma)[\sigma] \not\models_{DB} \Psi$ holding due to $\Psi \notin concl(\widehat{psec}_{prior}, prior)$.

5.2.4 Confidentiality Compromising Disjunctions

When considering confidentiality policies containing *multiple* single premise tuple generating dependencies, another harmful inference-channel may be provided, if the conclusions of different dependencies imply the *same* potential secret of a confidentiality policy. For instance, consider an adversary's a priori knowledge

$$prior = \{\ \Gamma_1 = R(a,b,c) \Rightarrow (\exists Y)\, R(g,h,Y),\quad \Gamma_2 = R(b,b,c) \Rightarrow (\exists Y)\, R(g,Y,h)\ \}$$

and further suppose that the extended and cleaned confidentiality policy is

$$\widehat{psec}_{prior} = \{\, R(a,b,c),\ R(b,b,c),\ (\exists X_1)(\exists X_2)\, R(g,X_1,X_2)\,\}\ .$$

As $R(a,b,c)$ and $R(b,b,c)$ are *not* in the set $concl(\widehat{psec}_{prior}, prior)$ of those potential secrets, which are to be refused because of being implied by the conclusion of a dependency under an arbitrary constant substitution, the disjunction

$$R(a,b,c) \vee R(b,b,c)$$

might be constructed to weaken the knowledge about the satisfaction of (at least) one of the potential secrets $R(a,b,c)$ and $R(b,b,c)$ by an original database instance. Then, the adversary exploiting his a priori knowledge knows that this original instance must satisfy at least one of the conclusions $(\exists Y)\, R(g,h,Y)$ and $(\exists Y)\, R(g,Y,h)$, too, as at least one of the premises of the dependencies Γ_1 and Γ_2 is satisfied by this original instance, which is known to satisfy the disjunction $R(a,b,c) \vee R(b,b,c)$. So, because of both

- $(\exists Y)\, R(g,h,Y) \models_{DB} (\exists X_1)(\exists X_2)\, R(g,X_1,X_2)$ and
- $(\exists Y)\, R(g,Y,h) \models_{DB} (\exists X_1)(\exists X_2)\, R(g,X_1,X_2)$,

the adversary can conclude that the potential secret $(\exists X_1)(\exists X_2)\, R(g,X_1,X_2)$ is also satisfied by the original instance, thereby violating the confidentiality policy.

More generally, this kind of inference-channel occurs, if there is a subset of single premise tuple generating dependencies, whose conclusions all imply the same potential secret Ψ (under certain constant substitutions), and each disjunct of a disjunction revealed to an adversary implies the validity of a premise of one of these dependencies (under the considered constant substitutions). Then, an alternative instance obeying the potential secret Ψ can *not* be constructed, as this alternative instance needs to satisfy this disjunction, which implies the validity of Ψ when also considering the adversary's a priori knowledge.

This kind of inference-channel can also occur in combination with transitive chains of dependencies. Then, the conclusion of a dependency, whose premise is satisfied (directly or transitively) by a disjunct under a certain constant substitution, implies the satisfaction of the premise of another dependency under this constant substitution. For instance, suppose that the above given example is modified such that the adversary's a priori knowledge now is

$$\begin{aligned} prior = \{\ \Gamma_1 &= R(b,a,a) \Rightarrow R(c,b,b), \\ \Gamma_2 &= (\forall X)\,[\, R(X,b,b) \Rightarrow R(a,b,X)\,], \\ \Gamma_3 &= R(a,b,c) \Rightarrow (\exists Y)\, R(g,h,Y), \\ \Gamma_4 &= R(b,b,a) \Rightarrow (\exists Y)\, R(g,Y,h) \qquad \} \end{aligned}$$

and the extended and cleaned confidentiality policy $\widehat{psec}_{prior}$ now contains (among others) the potential secrets

$$R(b,a,a),\ R(b,b,a) \text{ and } (\exists X_1)(\exists X_2)\, R(g,X_1,X_2)\ .$$

Then, the disjunction $R(b,a,a) \vee R(b,b,a)$ might be constructed to weaken the knowledge about the satisfaction of (at least) one of the potential secrets $R(b,a,a)$ and $R(b,b,a)$ by a considered original database instance and implies the potential secret $(\exists X_1)(\exists X_2)\, R(g,X_1,X_2)$:

- under the assumption that $R(b, a, a)$ is satisfied by an alternative instance, the transitive chain of the dependencies Γ_1, Γ_2 and Γ_3 requires that the conclusion of Γ_3 must be satisfied, i.e., the sentence $(\exists Y)\, R(g, h, Y)$ implying the potential secret $(\exists X_1)(\exists X_2)\, R(g, X_1, X_2)$;
- if $R(b, b, a)$ is instead satisfied by an alternative instance, the conclusion of Γ_4 must be satisfied, i.e., the sentence $(\exists Y)\, R(g, Y, h)$ also implying the potential secret $(\exists X_1)(\exists X_2)\, R(g, X_1, X_2)$.

To prevent the construction of disjunctions, which are confidentiality compromising in combination with an adversary's a priori knowledge *prior*, a disjunction must *not* only consist of disjuncts all implying the satisfaction of the same potential secret with the help of dependencies of *prior*. In a scenario *without* a priori knowledge, there is *no* pair of disjuncts implying the same potential secret: each disjunct is a potential secret stemming from a *cleaned* confidentiality policy guaranteeing the isolation property that there is *no* implication relationship between each pair of different potential secrets of this policy. But dependencies of *prior* interfering with potential secrets might break this isolation up by bridging the gaps between pairs of different potential secrets of a cleaned confidentiality policy and might thereby allow for reasoning about the satisfaction of one potential secret based on the satisfaction of (a set of) other potential secrets.

When neglecting transitive chains of dependencies for now, a sufficient idea to identify those disjunction templates constructed over an extended and cleaned confidentiality policy $\widehat{psec}_{prior}$, which are possibly harmful in combination with an adversary's a priori knowledge *prior*, is to partition the dependencies of *prior* with respect to $\widehat{psec}_{prior}$ as follows: each pair $\Gamma, \bar{\Gamma} \in prior$ is to be in the same partition P_i of *prior*, if there is a potential secret Ψ of $\widehat{psec}_{prior}$ implied by the conclusions of both of these dependencies Γ and $\bar{\Gamma}$ under arbitrary constant substitutions. Then, a disjunction template Φ is possibly harmful, if there is a single partition P_i of dependencies such that *each* of the disjuncts of Φ implies a premise of a dependency of P_i. If the construction of these possibly harmful disjunction templates is avoided, an additional isolation property guaranteeing that for each potential secret $\Psi \in \widehat{psec}_{prior}$ each constructed disjunction template contains at least one disjunct *not* implying Ψ is established.

When also considering transitive chains of dependencies, the partitioning of *prior* must moreover reflect that an adversary might infer the satisfaction of a potential secret by employing such a transitive chain. A sufficient condition to achieve this is to additionally guarantee that a pair $\Gamma, \bar{\Gamma}$ of dependencies of *prior* is also in the same partition P_i, if the conclusion of Γ implies the premise of $\bar{\Gamma}$ under an arbitrary constant substitution. As a consequence, each pair of possible transitive chains of

dependencies both implying the same potential secret is completely contained in one partition of dependencies.

Summing up the ideas discussed above, a partitioning of an adversary's a priori knowledge *prior* allowing for the detection of disjunction templates, which are possibly harmful in combination with *prior*, is defined as follows:

Definition 5.7: Partitioning of A Priori Knowledge

Let *prior* be an adversary's a priori knowledge consisting of single premise tuple generating dependencies and let $\widehat{psec}_{prior}$ be a confidentiality policy, which has first been extended with respect to *prior* and then been cleaned.

Then, $\mathcal{P}$ is a *partitioning* of *prior* with respect to $\widehat{psec}_{prior}$, if

(i) $\mathcal{P} := P_1 \dot{\cup} \ldots \dot{\cup} P_q$ is an ordinary partitioning of *prior*, i.e.,

  (a) $P_i \neq \emptyset$ and $P_i \subseteq prior$ for each $P_i \in \mathcal{P}$,

  (b) $P_i \cap P_j = \emptyset$ for all pairs of different partitions $P_i, P_j \in \mathcal{P}$,

  (c) $\bigcup_{P_i \in \mathcal{P}} P_i = prior$,

(ii) for each pair of different dependencies $\Gamma, \bar{\Gamma} \in prior$, for which there is a potential secret $\Psi \in \widehat{psec}_{prior}$ with both $concl(\Gamma)[\sigma] \models_{DB} \Psi$ and $concl(\bar{\Gamma})[\bar{\sigma}] \models_{DB} \Psi$ under arbitrary constant substitutions σ and $\bar{\sigma}$, there is a *single* partition $P_i \in \mathcal{P}$ with both $\Gamma \in P_i$ and $\bar{\Gamma} \in P_i$,

(iii) for each pair of different dependencies $\Gamma, \bar{\Gamma} \in prior$, for which the implication $concl(\Gamma)[\sigma] \models_{DB} prem(\bar{\Gamma})$ holds under an arbitrary constant substitution σ, there is a *single* partition $P_i \in \mathcal{P}$ with both $\Gamma \in P_i$ and $\bar{\Gamma} \in P_i$, and

(iv) q (and hence the number of partitions) is maximum.

Without the additional requirement that the number of partitions is to be maximized, a trivial but feasible partitioning can always be found by constructing only one single partition containing each dependency of *prior*. Of course, such a solution should be avoided (if possible), as it might unnecessarily forbid the construction of admissible disjunction templates, which can actually *not* enable an adversary exploiting his a priori knowledge to draw harmful inferences.

One way to algorithmically determine a partitioning $\mathcal{P}$ of an adversary's a priori knowledge *prior* is to first create a so-called partitioning graph in the form of

an undirected graph $G = (V, E)$, whose set V of vertices contains exactly one vertex v_Γ for each dependency $\Gamma \in \mathit{prior}$. Two different vertices $v_{\Gamma_1}, v_{\Gamma_2} \in V$ are then neighbored by introducing the (undirected) edge $\{v_{\Gamma_1}, v_{\Gamma_2}\} \in E$, *if and only if* their corresponding dependencies Γ_1 and Γ_2 need to be in the same partition according to requirement (ii) or requirement (iii) of Definition 5.7. After that, the wanted partitioning $\mathcal{P}$ of *prior* can be determined by decomposing the graph G into its connected components [65, 67]. Then, for each connected component $C_i = \{v_{\Gamma_{i_1}}, \ldots, v_{\Gamma_{i_k}}\}$ of G the corresponding partition $P_i = \{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$ of dependencies of *prior* is added to the initially empty partitioning $\mathcal{P}$.

As different connected components of a graph are always pairwise vertex-disjoint and as the constructed graph is supposed to contain exactly one vertex v_Γ for each dependency $\Gamma \in \mathit{prior}$, the algorithm sketched above is guaranteed to construct an ordinary partitioning of *prior* and thereby satisfies requirement (i) of Definition 5.7. Moreover, the definition of connected components guarantees that neighbored vertices of a graph are in the same connected component. The constructed partitioning $\mathcal{P}$ hence satisfies the requirements (ii) and (iii) of Definition 5.7 due to the construction of the edges of a partitioning graph.

Under the assumption that the algorithm returns a set $\mathcal{P}$ of partitions, which is *not* maximum according to requirement (iv) of Definition 5.7, there is at least one non-singleton partition $P_i = \{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$, which can be safely split up into two partitions P_{i_1} and P_{i_2} with $P_{i_1} \neq \emptyset$ and $P_{i_2} \neq \emptyset$ and with $P_{i_1} \dot\cup P_{i_2} = P_i$ without violating any of the requirements (i), (ii) and (iii) of Definition 5.7. By construction of the algorithm, the partitioning graph G then contains the connected component $C_i = \{v_{\Gamma_{i_1}}, \ldots, v_{\Gamma_{i_k}}\}$ inducing the considered partition P_i. But due to the definition of connected components, the partitioning graph G contains a path between each pair of vertices of C_i and according to the construction of the edges of G all dependencies corresponding to the vertices of C_i need to be in the same partition according the requirements (ii) and (iii) to Definition 5.7.

5.2.5 Reconsidering the Construction of Clusterings

Now that disjunction templates, which might possibly be harmful in combination with an adversary's a priori knowledge, can be identified, care must be taken to ensure that all disjunction templates induced by a clustering are *not* possibly harmful. As proposed in Section 4.1, a clustering with clusters of size 2 can be constructed for the availability-maximizing weakening approach on the basis of a maximum matching computed on an indistinguishability graph, each of whose edges represents an admissible cluster, which might be chosen for the final disjoint clustering. To prevent such a maximum matching algorithm from choosing

clusters, which correspond to possibly harmful disjunction templates, each admissible edge is from now on only added to an indistinguishability graph, if its corresponding disjunction template is *not* classified as possibly harmful.

Reconsidering Section 5.2.3, disjunction templates must further *not* contain any potential secrets, which need to be refused completely because of being implied by the conclusion of a dependency of an adversary's a priori knowledge. To actually prevent a maximum matching algorithm from constructing clusters containing these potential secrets to be refused, the set of vertices of an indistinguishability graph is now restricted to those potential secrets of an extended and cleaned confidentiality policy $\widehat{psec}_{prior}$, which are *not* to be refused.

Definition 5.8: Indistinguishability Graph (Extended)

Let $\widehat{psec}_{prior}$ be a confidentiality policy, which has first been extended with respect to an adversary's a priori knowledge *prior* of single premise tuple generating dependencies and which has then been cleaned. Moreover, let $\mathcal{P}$ be the partitioning of *prior* with respect to $\widehat{psec}_{prior}$ and let $concl(\widehat{psec}_{prior}, prior)$ be the subset of those potential secrets of $\widehat{psec}_{prior}$, which are to be refused. Finally, suppose that a notion of admissible indistinguishabilities is given, which induces a set $\mathcal{C}^a$ of admissible clusters of size 2 over $\widehat{psec}_{prior}$.

An indistinguishability graph extended for dealing with single premise tuple generating dependencies is an undirected graph $G = (V, E)$ such that

(i) $V := \widehat{psec}_{prior} \setminus concl(\widehat{psec}_{prior}, prior)$ is the set of vertices of G and

(ii) each (unordered) pair $\{\Psi_1, \Psi_2\}$ with $\Psi_1, \Psi_2 \in V$ and $\Psi_1 \neq \Psi_2$ constitutes an undirected edge of E, if

  (a) $\{\Psi_1, \Psi_2\}$ is in the set $\mathcal{C}^a$ of admissible clusters over $\widehat{psec}_{prior}$ and

  (b) there is *no* single partition $P_i \in \mathcal{P}$ containing dependencies $\Gamma_1 \in P_i$ and $\Gamma_2 \in P_i$ (with possibly $\Gamma_1 = \Gamma_2$) such that both implications $\Psi_1 \models_{DB} prem(\Gamma_1)$ and $\Psi_2 \models_{DB} prem(\Gamma_2)$ hold.

Although the computation of a maximum matching on an indistinguishability graph is a straightforward approach to determine a disjoint clustering inducing a set of pairwise disjoint disjunction templates of length 2, it is known from Section 4.1.2 that such a maximum matching does *not* necessarily cover each element of a given confidentiality policy. Within the weakening approach *not* handling a priori knowledge, which only allows for the enforcement of policy elements with

the help of suitable weakening disjunctions, this observation immediately leads to the need of constructing a so-called matching extension in the sense of Definition 4.3 – which essentially requires that each uncovered policy element must be paired with an (artificial) additional potential secret – to guarantee that there is always such a suitable weakening disjunction for each policy element.

But within the extended weakening approach, which is able to handle single premise tuple generating dependencies as a priori knowledge, there is actually more freedom to distort confidential knowledge: as known from Section 5.2.3, there may be the need to refuse the knowledge embodied in certain potential secrets completely to be able to guarantee the constructibility of credible alternative instances. So, in contrast to the non-extended weakening approach, there is *no* inevitable need to weaken all confidential knowledge with the help of suitable disjunctions and thus potential secrets uncovered by a maximum matching can in principle also be refused by the weakening algorithm.

As a consequence of these extended possibilities to distort confidential knowledge, employed notions of admissible indistinguishabilities do *not* necessarily need to be well-defined. Within the non-extended approach, which is only capable of weakening knowledge, the well-definedness of these notions is of crucial importance to guarantee that an admissible disjunction template can actually be determined for each element of a confidentiality policy despite the restrictions such a notion imposes on the construction of admissible disjunction templates. But within the extended approach, which can employ both weakening disjunctions and refusals, those potential secrets, for which a considered notion of admissible indistinguishabilities does *not* allow for the construction of a suitable disjunction template, can still be enforced by refusing them – although the weakening of confidential knowledge should still be the method of choice in terms of availability.

As a further consequence of these possible refusals of uncovered policy elements, the notion of a matching extension given in Definition 4.3 needs to be relaxed. While this notion obviously makes sense for the non-extended weakening approach to guarantee the existence of a weakening disjunction for each policy element, its requirement that each policy element should be in an (extended) cluster of size 2 does obviously *not* make sense for potential secrets enforced by refusals. Such a relaxed notion of a matching extension is in the following referred to as a *partly extended matching*.

Similar to Definition 4.3 of a matching extension, the following definition of a partly extended matching also relies on an extended clustering according to Definition 3.3. But in contrast to a matching extension M^*, which is supposed to be an extended clustering of all potential secrets of a given confidentiality policy, a partly extended matching M^+ only needs to be an extended clustering of the

subset of those potential secrets of a given confidentiality policy, which actually occur in a cluster of M^+. The remaining policy elements, which are *not* covered by this partly extended matching M^+, do instead *not* need to be a part of this extended clustering.

Definition 5.9: Partly Extended Matching

Let $\widehat{psec}_{prior}$ be an extended and cleaned confidentiality policy and suppose that M is a maximum matching on the indistinguishability graph corresponding to the set of admissible clusters over $\widehat{psec}_{prior}$, which is induced by a given notion of admissible indistinguishabilities.

A *partly extended matching* M^+ of the maximum matching M and the policy $\widehat{psec}_{prior}$ under the given notion of admissible indistinguishabilities

(i) contains each matching edge of M, i.e., $M \subseteq M^+$, and

(ii) can possibly contain a cluster $\{\Psi, \Psi^A\} \in M^+$ for each potential secret $\Psi \in \widehat{psec}_{prior}$ not covered by M such that

- Ψ^A is an additional potential secret not occurring in $\widehat{psec}_{prior}$ and
- M^+ is an extended clustering of the set $\bigcup_{\{\Psi_1, \Psi_2\} \in M^+} \{\Psi_1, \Psi_2\}$ of all (possibly additional) potential secrets of M^+ with clusters of size 2 according to Definition 3.3, which obeys the given notion of admissible indistinguishabilities.

Moreover, such a partly extended matching M^+ covers a potential secret $\Psi \in \widehat{psec}_{prior}$, if there is a cluster $\{\Psi_1, \Psi_2\} \in M^+$ with $\Psi \in \{\Psi_1, \Psi_2\}$.

Although this definition of a partly extended clustering does *not* explicitly require that potential secrets uncovered by a computed maximum matching should be enforced with the help of a corresponding weakening disjunction whenever possible, the construction goals of the weakening algorithm postulated in Section 1.3 – aiming at cooperativeness in terms of availability – suggest to do so. As this results in the construction of additional potential secrets, care must be taken to ensure that the set of all constructed additional potential secrets does *not* enable an adversary to compromise the confidentiality policy.

Hence, similar to the discussion in Section 3.1.2 that the sets of all non-additional potential secrets and of all additional potential secrets should together form a cleaned set to preserve the isolation property of non-implication, the set of all additional potential secrets of a partly extended clustering should *not* violate any

isolation property of the extended weakening algorithm. The resulting requirements for an admissible set of additional potential secrets are captured in the following definition, which (similar to Definition 3.5) implements the purely generic requirement an extended clustering (and hence also a partly extended matching) has to satisfy according to condition (v) of Definition 3.3.

Definition 5.10: Admissible Additional Potential Secrets (Ext.)

Consider an adversary's a priori knowledge *prior* consisting of single premise tuple generating dependencies and a confidentiality policy $\widehat{psec}_{prior}$, which is extended with respect to *prior* and which is cleaned.

A set $\widehat{psec}^{A}_{prior}$ of additional potential secrets is *admissible* with respect to $\widehat{psec}_{prior}$ and *prior*, if

(i) the union $\widehat{psec}_{prior} \cup \widehat{psec}^{A}_{prior}$ is a cleaned set,

(ii) *prior* does *not* interfere with $\widehat{psec}^{A}_{prior}$ and

(iii) the non-implication $\Gamma \not\models_{DB} \Psi^{A}$ holds for each dependency $\Gamma \in prior$ and each additional potential secret $\Psi^{A} \in \widehat{psec}^{A}_{prior}$.

Requirement (ii) of this definition guarantees that a constructed additional potential secret can occur in a disjunction and does *not* need to be refused completely because of being implied by the conclusion of dependency. Moreover, this requirement takes care that the construction of an additional potential secret does *not* lead to any new interferences between dependencies of the a priori knowledge and the extended policy possibly leading to the need to additionally refuse other potential secrets completely. As another consequence of this non-interference, which in particular requires that a constructed additional potential secret does *not* imply the premise of any dependency, each disjunction containing an additional potential secret contains at least one disjunct *not* enabling the adversary to conclude any (positive) knowledge with the help of his a priori knowledge.

5.3 Adapting the Construction of Weakened Views

Now that the clustering of confidentiality policies has been adapted to eliminate inference-channels, which might occur due to an adversary's a priori knowledge consisting of single premise tuple generating dependencies, the construction of a weakened view on a considered original database instance is again the next step

to follow. Similar to the construction of weakened views described in Section 3.2, a weakened view should again distort all confidential knowledge effectively. But, in contrast to the non-extended weakening algorithm *not* handling a priori knowledge – which enforces all specified confidentiality requirements solely on the basis of weakening disjunctions – a weakened view must now additionally be capable of refusing any knowledge about those potential secrets of a considered (extended and cleaned) confidentiality policy, which do *not* occur in disjunction templates induced by a partly extended matching for this policy.

Conceptually, weakened views as known from Definition 3.7 can be extended to be capable of refusing potential secrets by ensuring that a weakened view does *neither* reveal the satisfaction *nor* the non-satisfaction of a potential secret *not* occurring in any disjunction template. In particular, additional care has to be taken that *no* ground atom of the positive knowledge of a weakened view implies the satisfaction of such a potential secret to be refused and that the completeness sentence of a weakened view does *not* reveal the non-satisfaction of such a potential secret. Then, the validity status of a potential secret to be refused remains completely unknown, as the isolation properties established for extended and cleaned confidentiality policies further guarantee that the satisfaction or non-satisfaction of such a potential secret can *not* be concluded on the basis of weakening disjunctions or negated disjunctions revealed by a weakened view.

Following these ideas, the subset of all refused potential secrets is easy to identify from an adversary's point of view: an adversary is supposed to be aware of a considered extended and cleaned confidentiality policy and each potential secret of such a policy, which does *neither* occur in a weakening disjunction *nor* in a negated disjunction, is known to be refused according to the construction of weakened views. However, following the design guidelines developed in Section 1.3, it seems worthwhile to list all refused policy elements explicitly within a weakened view to make these distortions readily identifiable. But then, the way these sentences of the refused knowledge of a weakened view are to be interpreted differs from all other sentences of such a weakened view: while each sentence occurring in the positive knowledge, the disjunctive knowledge or the negative knowledge of a weakened view is known to be satisfied by a considered original database instance (cf. Section 3.2), a weakened view does *not* reveal any information about the validity status of each sentence listed as refused knowledge.

These insights lead to the following construction of weakened views, which are extended to be also capable of handling refused knowledge. Thereby, a weakened view is again supposed to consist of ordered sequences of sentences to prevent an adversary from drawing harmful meta-inferences on the basis of the syntactic appearance of weakened views.

Definition 5.11: Weakened View (Possibly with Refusals)

Let r be a complete database instance over a database schema $\langle R|\mathcal{A}_R|SC_R\rangle$ with SC_R being a subset of an adversary's a priori knowledge *prior* consisting of single premise tuple generating dependencies. Further, suppose that M_r^+ is the subset of those clusters of a partly extended matching M^+, which is constructed for an extended and cleaned confidentiality policy $\widehat{psec}_{prior}$, such that $\mathcal{I}_r \models_M \bigvee_{\Psi\in C} \Psi$ holds for each cluster $C \in M_r^+$.

Then, the *weakened view* $weak(r, psec, prior)$ on r consists of the following totally ordered sequences of sentences of the first-order language $\mathscr{L}$:

(i) *Positive knowledge* $weak(r, psec, prior)^+$: Each tuple $\boldsymbol{c} \in r$, for which the non-implication $R(\boldsymbol{c}) \not\models_{DB} \Psi$ holds for

- each (possibly additional) potential secret $\Psi \in \bigcup_{C\in M_r^+} C$ and
- each potential secret $\Psi \in \left(\widehat{psec}_{prior} \setminus \bigcup_{C\in M^+} C\right)$ not occurring in any cluster of M^+ (and hence to be refused),

is modeled as the ground atom $R(\boldsymbol{c})$.

(ii) *Disjunctive knowledge* $weak(r, psec, prior)^\vee$: For each cluster $C \in M_r^+$ the disjunction $\bigvee_{\Psi\in C} \Psi$ is constructed.

(iii) *Refused knowledge* $weak(r, psec, prior)^?$: For each potential secret $\Psi \in \left(\widehat{psec}_{prior} \setminus \bigcup_{C\in M^+} C\right)$ the sentence Ψ is marked as refused knowledge.

(iv) *Negative knowledge* $weak(r, psec, prior)^-$: For each $C \in (M^+ \setminus M_r^+)$ the negated disjunction $\neg[\bigvee_{\Psi\in C} \Psi]$ is constructed. Moreover, a (partial) completeness sentence having a universally quantified variable X_j for each attribute $A_j \in \mathcal{A}_R$ is built. This sentence is supposed to contain 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 $weak(r, psec, prior)^+$,
- each sentence $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ of $weak(r, psec, prior)^?$ and
- each disjunct $(\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ occurring in $weak(r, psec, prior)^\vee$

and finally contains $\neg R(X_1, \ldots, X_n)$ as its last disjunct.

Under the supposition that there is a total order on the set of those (constant and variable) symbols, which might appear as terms of sentences of a weakened view, a total order on the sentences actually occurring within a weakened view

$weak(r, psec, prior)$ can for example – by suitably adapting the ideas proposed in Section 3.2 – be established as follows:

- the presentation of the weakened view starts with all ground atoms of the positive knowledge $weak(r, psec, prior)^+$ and the sequence of these ground atoms is sorted lexicographically according to the order on their constant symbols;

- then, all disjunctions of the disjunctive knowledge $weak(r, psec, prior)^\vee$ follow such that first within each of these disjunctions the sequence of its disjuncts is sorted lexicographically according to the order on their terms and subsequently the sequence of all of these disjunctions is sorted lexicographically according to the order on their terms;

- after that all potential secrets of $weak(r, psec, prior)^?$ are exposed as refused knowledge and the sequence of these sentences is sorted lexicographically according to the order on their terms;

- subsequently all negated disjunctions occurring in the negative knowledge $weak(r, psec, prior)^-$ are presented and the sequence of (and within) these sentences is ordered just as the disjunctions of $weak(r, psec, prior)^\vee$;

- and finally, the (partial) completeness sentence of the negative knowledge $weak(r, psec, prior)^-$ is given and normalized as follows: first, within each of its disjuncts of the form $(\bigwedge_{i \in \{1,\ldots,n\} \text{ with } t_i \in Dom} X_i \equiv t_i)$ the sequence of the conjuncts of the form $X_i \equiv t_i$ is sorted lexicographically according to the order on the variable symbols X_i and after that the sequence of these disjuncts is sorted lexicographically according to the order on their terms; then, $\neg R(X_1, \ldots, X_n)$ is finally appended as the last disjunct.

Now that all basic subroutines needed for the construction of an extended weakening algorithm, which is capable of handling an adversary's a priori knowledge in the form of single premise tuple generating dependencies, are developed, the overall extended weakening algorithm can be specified. Thereby, all of these basic subroutines, except for the construction of an indistinguishability graph and the computation of a partly extended matching, whose algorithmic instantiation crucially depend on the employed notion of admissible indistinguishabilities, are specified on the operational level. Similar to Section 4.1.3, the developed weakening algorithm is availability-maximizing in the sense that it aims at the construction of weakening disjunctions of length 2, which are the shortest possible non-trivial disjunctions and guarantee the existence of only one secure alternative instance for both of their elements.

Algorithm 5.1: Inference-Proof Weakening (Extended)

Let r be a complete database instance over a database schema $\langle R|\mathcal{A}_R|SC_R\rangle$, let *psec* be a confidentiality policy of existentially quantified atoms and suppose that a notion of admissible indistinguishabilities is given. Moreover, assume that an adversary's a priori knowledge *prior* (with $SC_R \subseteq prior$) consisting of single premise tuple generating dependencies is given such that $\mathcal{I}_r \models_M prior$ holds and $prior \not\models_{DB} \Psi$ is valid for each $\Psi \in psec$.

Then, a weakened view $weak(r, psec, prior)$ on r is created as follows:

- **Stage 1** (independent of r): Disjoint clustering of potential secrets
  - (i) Create the extended confidentiality policy $psec_{prior}$ (Def. 5.5)
  - (ii) Construct the cleaned set $\widehat{psec}_{prior}$ based on $psec_{prior}$ (Def. 3.4)
  - (iii) Identify the complete refusals of $concl(\widehat{psec}_{prior}, prior)$ (Def. 5.6)
  - (iv) Determine the partitioning $\mathcal{P}$ of *prior* w.r.t. $\widehat{psec}_{prior}$ (Def. 5.7)
  - (v) Generate the indistinguishability graph $G = (V, E)$ (Def. 5.8)
  - (vi) Compute a maximum matching M on G (Def. 4.2)
  - (vii) Create a partly extended matching M^+ (Def. 5.9)
- **Stage 2** (dependent on r): Creation of weakened view
  - (viii) Create the subset $M_r^+ := \{\, C \in M^+ \mid \mathcal{I}_r \models_M \bigvee_{\Psi \in C} \Psi \,\}$ of clusters containing a potential secret satisfied by $\mathcal{I}_r$
  - (ix) Create the weakened view $weak(r, psec, prior)$ on r (Def. 5.11)

As already mentioned in Section 5.1, it might be worthwhile to require that at least one term of the premise and at least one term of the conclusion of each considered single premise tuple generating dependency of an adversary's a priori knowledge is a constant symbol of *Dom*. Otherwise, there might be a dependency Γ in an adversary's a priori knowledge *prior* having an existentially quantified premise $prem(\Gamma)$ or an existentially quantified conclusion $concl(\Gamma)$, which is semantically equivalent to the weakest possible existentially quantified atom of the form $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$. This dependency Γ then interferes with each non-empty confidentiality policy *psec*: if $prem(\Gamma)$ is semantically equivalent to $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$, the implication $\Psi \models_{DB} prem(\Gamma)$ holds for each potential secret $\Psi \in psec$, and if $concl(\Gamma)$ is semantically equivalent to $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$, each $\Psi \in psec$ and $concl(\Gamma)$ share a common constant unifier.

Considering an arbitrary non-empty confidentiality policy *psec*, the corresponding extended confidentiality policy $psec_{prior}$ contains both $prem(\Gamma)$ and $concl(\Gamma)$ because of the above described interference and thus also a weakest possible existentially quantified atom of the form $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$. As each other element of the extended policy $psec_{prior}$ implies $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$, the corresponding cleaned confidentiality policy $\widehat{psec}_{prior}$ consists of solely one remaining element semantically equivalent to $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$, which is moreover implied by $concl(\Gamma)$. This single remaining potential secret of the form $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$ is hence to be refused completely and, as an immediate consequence, this leads to a complete refusal of all knowledge about a considered original database instance.

Of course, this worst-case scenario in terms of availability also occurs, if a weakest possible potential secret of the form $(\exists \boldsymbol{X})\, R(\boldsymbol{X})$ is contained in a considered confidentiality policy *psec*. It hence makes sense to avoid the construction of confidentiality policies containing such a potential secret whenever possible, although such policies are – in contrast to the non-extended weakening approach enforcing confidentiality policies solely on the basis of weakening disjunctions – perfectly decent from a formal point of view.

To now give an overall example of the extended availability-maximizing instantiation of the weakening algorithm, consider the input instances given in Figure 5.1. Thereby, both the original database instance r of Figure 5.1(a) and the confidentiality policy *psec* of Figure 5.1(c) are taken over from the example of the (non-extended) weakening algorithm known from Figure 4.4 and are further complemented by the a priori knowledge *prior* given in Figure 5.1(b). One can easily see that the considered database instance r complies with the considered a priori knowledge *prior*, as the DB-Interpretation $\mathcal{I}_r$ induced by r satisfies *prior*.

Within the first step of Stage 1 of Algorithm 5.1 the given confidentiality policy *psec* is extended with respect to the given a priori knowledge *prior*. Thereby, the dependencies Γ_1, Γ_2, Γ_3 and Γ_5 immediately interfere with *psec*, leading to the (partly) extended confidentiality policy

$$\begin{aligned} psec_{prior} = psec \cup \{\ & (\exists X)\, R(a, X, c),\ (\exists X)\, R(X, d, f),\ (\exists X)\, R(X, a, e), \\ & (\exists X)\, R(a, X, d),\ (\exists X)\, R(X, b, e),\ (\exists X)\, R(a, X, a), \\ & (\exists X)(\exists Y)\, R(X, e, Y) \qquad\qquad \}\ . \end{aligned}$$

After this extension of the policy, the dependency Γ_4 now also interferes with $psec_{prior}$ because of $concl(\Gamma_4) = (\exists X)\, R(g, e, X)$ sharing a common constant unifier with $(\exists X)(\exists Y)\, R(X, e, Y) \in psec_{prior}$. As a consequence, the (already partly)

$$r = \{\, (a,b,c),\ (a,f,g),\ (b,a,e),\ (b,b,d),\ (b,d,f),\ (g,e,i),\ (g,h,i) \,\}$$

(a) Original database instance r (complying with *prior*)

$$\begin{aligned} prior = \{\ \ \Gamma_1 &= (\forall X)\,[\,R(a,X,c) \Rightarrow R(X,d,f)\,] \\ \Gamma_2 &= (\forall X)\,[\,R(X,d,f) \Rightarrow R(X,a,e)\,] \\ \Gamma_3 &= (\forall X)\,[\,R(a,X,d) \Rightarrow R(X,b,e)\,] \\ \Gamma_4 &= (\forall X)\,[\,R(g,h,X) \Rightarrow R(g,e,X)\,] \\ \Gamma_5 &= (\forall X)\,[\,R(a,X,a) \Rightarrow (\exists Y)\,R(X,e,Y)\,]\ \ \} \end{aligned}$$

(b) Adversary's a priori knowledge *prior*

$$\begin{aligned} psec = \{\ & R(a,a,a),\ R(a,b,a),\ R(a,b,c),\ R(a,b,d),\ R(a,b,e),\ R(a,c,a), \\ & (\exists X)\,R(a,e,X),\ (\exists X)\,R(b,e,X),\ (\exists X)\,R(c,e,X),\ (\exists X)\,R(b,X,e)\ \} \end{aligned}$$

(c) Confidentiality policy *psec*

Figure 5.1: Possible input instances for Algorithm 5.1

extended policy $psec_{prior}$ is again extended to

$$\begin{aligned} psec_{prior} = psec \cup \{\ & (\exists X)\,R(a,X,c),\ (\exists X)\,R(X,d,f),\ (\exists X)\,R(X,a,e), \\ & (\exists X)\,R(a,X,d),\ (\exists X)\,R(X,b,e),\ (\exists X)\,R(a,X,a), \\ & (\exists X)(\exists Y)\,R(X,e,Y) \qquad\qquad\qquad\qquad\quad \} \cup \\ \{\ & (\exists X)\,R(g,h,X),\ (\exists X)\,R(g,e,X) \qquad\qquad\qquad\quad \}\ . \end{aligned}$$

Now that the extension of the confidentiality policy is completed, this extended policy $psec_{prior}$ is cleaned within the next step of the weakening algorithm and hence reduced to the maximum "core" subset of its weakest sentences. During this process, all elements of the original policy *psec* except for the potential secret $(\exists X)\,R(b,X,e)$ are removed, as each of these sentences implies another (weaker) sentence of $psec_{prior}$. Moreover, the sentence $(\exists X)\,R(g,e,X)$ of $psec_{prior}$ is also removed because of $(\exists X)\,R(g,e,X) \models_{DB} (\exists X)(\exists Y)\,R(X,e,Y)$. This finally leads to the extended and cleaned confidentiality policy

$$\begin{aligned} \widehat{psec}_{prior} = \{\ & (\exists X)\,R(b,X,e),\ (\exists X)\,R(a,X,c),\ (\exists X)\,R(X,d,f), \\ & (\exists X)\,R(X,a,e),\ (\exists X)\,R(a,X,d),\ (\exists X)\,R(X,b,e), \\ & (\exists X)\,R(a,X,a),\ (\exists X)(\exists Y)\,R(X,e,Y),\ (\exists X)\,R(g,h,X)\ \}\ . \end{aligned}$$

To next determine the subset $concl(\widehat{psec}_{prior}, prior)$ of those potential secrets of $\widehat{psec}_{prior}$, which are to be refused completely, note that all elements of $\widehat{psec}_{prior}$ corresponding to an existentially quantified conclusion of a dependency of *prior*

always need to be refused. Further, the policy element $(\exists X)\, R(b, X, e)$ of the extended and cleaned policy $\widehat{psec}_{prior}$ also needs to be refused completely, as the implication $concl(\Gamma_2)[\sigma] \models_{DB} (\exists X)\, R(b, X, e)$ holds under each constant substitution σ with $\sigma(X) = b$. This finally results in

$$\begin{aligned} concl(\widehat{psec}_{prior}, prior) = \{\ & (\exists X)\, R(b, X, e),\ (\exists X)\, R(X, d, f), \\ & (\exists X)\, R(X, a, e),\ (\exists X)\, R(X, b, e), \\ & (\exists X)(\exists Y)\, R(X, e, Y) \qquad\qquad \}\ . \end{aligned}$$

As a next step, Stage 1 of the algorithm requires that the given a priori knowledge *prior* is partitioned with respect to $\widehat{psec}_{prior}$. Thereby, the dependencies Γ_1 and Γ_2 obviously need to be in the same partition, as the existentially quantified conclusion of Γ_1 implies the existentially quantified premise of Γ_2. Further, the dependency Γ_3 also needs to be in this partition, because there is the potential secret $(\exists X)\, R(b, X, e) \in \widehat{psec}_{prior}$, for which both implications

- $concl(\Gamma_2)[\sigma_2] \models_{DB} (\exists X)\, R(b, X, e)$ and
- $concl(\Gamma_3)[\sigma_3] \models_{DB} (\exists X)\, R(b, X, e)$

hold under constant substitutions σ_2 and σ_3 with $\sigma_2(X) = \sigma_3(X) = b$.

Similarly, the dependencies Γ_4 and Γ_5 need to be in the same partition, as the policy $\widehat{psec}_{prior}$ contains the potential secret $(\exists X)(\exists Y)\, R(X, e, Y)$ with both

- $concl(\Gamma_4)[\sigma_4] \models_{DB} (\exists X)(\exists Y)\, R(X, e, Y)$ and
- $concl(\Gamma_5)[\sigma_5] \models_{DB} (\exists X)(\exists Y)\, R(X, e, Y)$

under arbitrary constant substitutions σ_4 and σ_5.

As a consequence, the algorithm creates the partitioning $\mathcal{P} = \{P_1, P_2\}$ with

$$\begin{aligned} P_1 = \{\ \Gamma_1 &= (\forall X)\,[\, R(a, X, c) \Rightarrow R(X, d, f)\,], \\ \Gamma_2 &= (\forall X)\,[\, R(X, d, f) \Rightarrow R(X, a, e)\,], \\ \Gamma_3 &= (\forall X)\,[\, R(a, X, d) \Rightarrow R(X, b, e)\,]\ \} \qquad \text{and} \end{aligned}$$

$$\begin{aligned} P_2 = \{\ \Gamma_4 &= (\forall X)\,[\, R(g, h, X) \Rightarrow R(g, e, X)\,], \\ \Gamma_5 &= (\forall X)\,[\, R(a, X, a) \Rightarrow (\exists Y)\, R(X, e, Y)\,]\ \}\ . \end{aligned}$$

As the next step of Stage 1, the indistinguishability graph is constructed over the set $\widehat{psec}_{prior} \setminus concl(\widehat{psec}_{prior}, prior)$ of those potential secrets, which should be weakened by disjunctions whenever possible. Again employing interchangeability (cf. Definition 4.5) as the notion of admissible indistinguishabilities, which

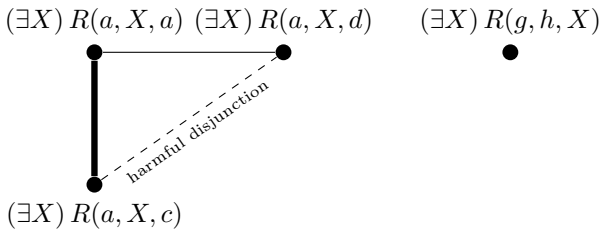


Figure 5.2: Indistinguishability graph with maximum matching

induces all edges of this graph, the resulting indistinguishability graph is depicted in Figure 5.2. Although the potential secrets $(\exists X)\, R(a, X, c)$ and $(\exists X)\, R(a, X, d)$ are obviously interchangeable, the indistinguishability graph does *not* contain an edge connecting the corresponding vertices of the graph, as both of these potential secrets imply an existentially quantified premise of the partition P_1 due to $\mathit{prem}\,(\Gamma_1) = (\exists X)\, R(a, X, c)$ and $\mathit{prem}\,(\Gamma_3) = (\exists X)\, R(a, X, d)$ and the disjunction $(\exists X)\, R(a, X, c) \vee (\exists X)\, R(a, X, d)$ might hence enable an adversary to compromise the confidentiality policy.

To actually compute a partly extended matching, the algorithm first determines a maximum matching

$$M = \{\, \{\, (\exists X)\, R(a, X, a),\ (\exists X)\, R(a, X, c)\,\}\,\}$$

on the considered indistinguishability graph. Then, this matching is partly extended to

$$\begin{aligned} M^+ = \{\, &\{\, (\exists X)\, R(a, X, a),\ (\exists X)\, R(a, X, c)\,\}, \\ &\{\, (\exists X)\, R(g, h, X),\ (\exists X)\, R(g, c, X)^A\,\}\,\} \end{aligned}$$

by pairing the uncovered potential secret $(\exists X)\, R(g, h, X)$ with the (admissible) additional potential secret $(\exists X)\, R(g, c, X)$.

Note that the potential secret $(\exists X)\, R(a, X, d)$ still remains uncovered by the partly extended matching M^+. Each interchangeable additional potential secret must differ from $(\exists X)\, R(a, X, d)$ either in the constant symbol at first position or in the constant symbol at the third position and in both of these cases the dependency Γ_5 of *prior* would interfere with the resulting additional potential secret due to a common constant unifier. Accordingly, an admissible additional potential secret can *not* be constructed for $(\exists X)\, R(a, X, d)$ and the only possibility to enforce this potential secret under the employed notion of interchangeability is to refuse it.

Now that the instance-independent Stage 1 of Algorithm 5.1 is completed, this algorithm continues with the instance-dependent Stage 2 finally returning the

$R(a, f, g)$
$R(b, b, d)$

$(\exists X)\, R(a, X, a) \vee (\exists X)\, R(a, X, c)$
$(\exists X)\, R(g, c, X) \vee (\exists X)\, R(g, h, X)$

Refused: { $(\exists X)\, R(X, a, e)$,
$(\exists X)\, R(X, b, e)$,
$(\exists X)\, R(X, d, f)$,
$(\exists X)(\exists Y)\, R(X, e, Y)$,
$(\exists X)\, R(a, X, d)$,
$(\exists X)\, R(b, X, e)$ }

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

Figure 5.3: Inference-proof weakened view for inputs of Figure 5.1

inference-proof weakened view given in Figure 5.3. Therefore, the algorithm first discovers that both disjunction templates corresponding to the extended clusters of M^+ are satisfied by the original database instance r given in Figure 5.1(a) and then constructs the weakened view as required by Definition 5.11.

5.4 Interchangeability Revisited

As demonstrated in the above given example of the extended weakening algorithm, the interchangeability criterion, which is introduced in Definition 4.5 as an example of a concrete notion of admissible indistinguishabilities, might *not* allow for the construction of an admissible additional potential secret for some potential secrets of a considered extended and cleaned confidentiality policy $\widehat{psec}_{prior}$. More specifically, an admissible additional potential secret can *not* be constructed for a potential secret $\Psi \in \widehat{psec}_{prior}$ according to Definition 5.10, if each additional potential secret for Ψ, which is constructible according to interchangeability, interferes with a dependency Γ of a considered adversary's a priori knowledge *prior* due to Ψ and $concl(\Gamma)$ sharing a common constant unifier.

The following theorem gives a both sufficient and necessary condition for the constructibility of additional potential secrets, which are both interchangeable and admissible. Thereby, the proof of this theorem is constructive in the sense that it sketches an efficient as well as easy to implement algorithm for the construction

of interchangeable and admissible additional potential secrets, provided that such an additional potential secret is constructible.

Theorem 5.1: Interchangeable Additional Potential Secrets

Consider an adversary's a priori knowledge *prior* consisting of single premise tuple generating dependencies and a confidentiality policy $\widehat{psec}_{prior}$, which is extended with respect to *prior* and which is cleaned. Moreover, suppose that $\widehat{psec}^{A}_{prior}$ is a set of additional potential secrets, which is *admissible* with respect to $\widehat{psec}_{prior}$ and *prior* according to Definition 5.10.

A further interchangeable additional potential secret Ψ^A can be constructed for a potential secret Ψ of $\widehat{psec}_{prior}$ such that the set $\widehat{psec}^{A}_{prior} \cup \{\Psi^A\}$ is still *admissible* with respect to $\widehat{psec}_{prior}$ and *prior*, *if and only if* there is a term t_m of $\Psi = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ with $t_m \in Dom$ such that for each dependency Γ of *prior* with $concl(\Gamma) = (\exists \boldsymbol{Y})\, R(t'_1, \ldots, t'_n)$

- the term t'_m of $concl(\Gamma)$ is a constant symbol, i.e., $t'_m \in Dom$, or
- Ψ and $concl(\Gamma)$ have at least one differing constant position different from m, i.e., there is an $i \in \{1, \ldots, n\} \setminus \{m\}$ such that both t_i and t'_i are constant symbols of *Dom* and $t_i \neq t'_i$ holds.

Proof. To start with the *only-if-part*, consider a potential secret Ψ of $\widehat{psec}_{prior}$ such that for each constant term t_m of $\Psi = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ with $t_m \in Dom$ there is a dependency Γ of *prior* with $concl(\Gamma) = (\exists \boldsymbol{Y})\, R(t'_1, \ldots, t'_n)$ and with both

- t'_m of $concl(\Gamma)$ being a variable of $\boldsymbol{Y}$ and
- $t_i = t'_i$ for each $i \in \{1, \ldots, n\} \setminus \{m\}$ such that both t_i and t'_i are constant symbols of *Dom*.

As $t_i = t'_i$ obviously also holds for each $i \in \{1, \ldots, n\}$, the existentially quantified atoms Ψ and $concl(\Gamma)$ share a common constant unifier according to Lemma 5.2 and as an immediate consequence the set $\widehat{psec}^{A}_{prior} \cup \{\Psi^A\}$ is *not* admissible according to Definition 5.10 due to *prior* interfering with $\widehat{psec}^{A}_{prior} \cup \{\Psi^A\}$.

To now continue with the *if-part*, consider a potential secret Ψ of $\widehat{psec}_{prior}$ having a constant term t_m of $\Psi = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ with $t_m \in Dom$ such that for each dependency Γ of *prior* with $concl(\Gamma) = (\exists \boldsymbol{Y})\, R(t'_1, \ldots, t'_n)$

- the term t'_m of $concl(\Gamma)$ is a constant symbol, i.e., $t'_m \in Dom$, or

- there is an $i \in \{1, \ldots, n\} \setminus \{m\}$ such that both t_i and t'_i are constant symbols of *Dom* and $t_i \neq t'_i$ holds.

Then, an interchangeable additional potential secret $\Psi^A = (\exists \boldsymbol{X})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ can be constructed for the potential secret $\Psi = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ as follows:

- set $\bar{t}_j := t_j$ for each $j \in \{1, \ldots, n\} \setminus \{m\}$ and
- choose $\bar{t}_m$ to be a constant symbol of *Dom* such that $\bar{t}_m \neq \tilde{t}_m$ holds for
  - each $(\exists \boldsymbol{Z})\, R(\tilde{t}_1, \ldots, \tilde{t}_n) \in \widehat{psec}_{prior} \cup \widehat{psec}^{A}_{prior}$ with $\tilde{t}_m \in Dom$,
  - each $concl(\Gamma) = (\exists \boldsymbol{Z})\, R(\tilde{t}_1, \ldots, \tilde{t}_n)$ of *prior* with $\tilde{t}_m \in Dom$ and
  - each $prem(\Gamma) = (\exists \boldsymbol{Z})\, R(\tilde{t}_1, \ldots, \tilde{t}_n)$ of *prior* with $\tilde{t}_m \in Dom$.

To start the proof that $\widehat{psec}^{A}_{prior} \cup \{\Psi^A\}$ is still admissible, first of all consider that *no* dependency $\Gamma \in prior$ interferes with $\Psi^A = (\exists \boldsymbol{X})\, R(\bar{t}_1, \ldots, \bar{t}_n)$, as required by condition (ii) of Definition 5.10. For that purpose, consider an arbitrary dependency Γ of *prior* with $concl(\Gamma) = (\exists \boldsymbol{Z})\, R(\tilde{t}_1, \ldots, \tilde{t}_n)$. Then, $\bar{t}_m \neq \tilde{t}_m$ is supposed to hold due to the construction of Ψ^A, if $\tilde{t}_m$ is a constant symbol of *Dom*. Otherwise, if $\tilde{t}_m$ is a variable, there is an $i \in \{1, \ldots, n\} \setminus \{m\}$ such that both t_i and t'_i are constant symbols of *Dom* and $t_i \neq t'_i$ holds. Hence, in both of these cases there is an $i \in \{1, \ldots, n\}$ such that both t_i and t'_i are constant symbols of *Dom* and $t_i \neq t'_i$ holds and as an immediate consequence Ψ^A and $concl(\Gamma)$ do *not* share a common constant unifier according to Lemma 5.2.

To now show that there is also *no* interference due to $\Psi^A \not\models_{DB} prem(\Gamma)$, assume that $\Psi^A \models_{DB} prem(\Gamma)$ holds for $prem(\Gamma) = (\exists \boldsymbol{Z})\, R(\tilde{t}_1, \ldots, \tilde{t}_n)$. As $\bar{t}_m \neq \tilde{t}_m$ is supposed to hold according to the construction of $\Psi^A = (\exists \boldsymbol{X})\, R(\bar{t}_1, \ldots, \bar{t}_n)$, the implication $\Psi^A \models_{DB} prem(\Gamma)$ can only hold, if the term $\tilde{t}_m$ of $prem(\Gamma)$ is a variable of $\boldsymbol{Z}$. But then, the potential secret $\Psi = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ of $\widehat{psec}_{prior}$, for which Ψ^A is constructed as an additional potential secret, also implies $prem(\Gamma)$, i.e., $\Psi \models_{DB} prem(\Gamma)$, as $\bar{t}_j = t_j$ holds according to the constructing of Ψ^A for each $j \in \{1, \ldots, n\} \setminus \{m\}$. Reconsidering that $\widehat{psec}_{prior}$ is supposed to be extended and cleaned, this interference between Γ and $\widehat{psec}_{prior}$ due to $\Psi \models_{DB} prem(\Gamma)$ guarantees that there is a potential secret $\Psi_{\text{prem}} \in \widehat{psec}_{prior}$ with $prem(\Gamma) \models_{DB} \Psi_{\text{prem}}$, leading to

$$\Psi \models_{DB} prem(\Gamma) \models_{DB} \Psi_{\text{prem}}$$

by transitivity. But then, Ψ and Ψ_{prem} – and hence also Ψ and $prem(\Gamma)$ – must be semantically equivalent: otherwise, it is *not* possible that both Ψ and Ψ_{prem} are in $\widehat{psec}_{prior}$ due to $\widehat{psec}_{prior}$ being a cleaned set. However, this semantic equivalence between $\Psi = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and $prem(\Gamma) = (\exists \boldsymbol{Z})\, R(\tilde{t}_1, \ldots, \tilde{t}_n)$ is obviously

in contradiction to the assumption that t_m is a constant symbol of *Dom* and that $\tilde{t}_m$ is a variable of $\boldsymbol{Y}$ and accordingly $\Psi^A \not\models_{DB} prem(\Gamma)$ holds. Moreover, considering that an interference relationship according to case (ii) of Definition 5.4 is *not* possible, as Ψ^A is *not* an existentially quantified conclusion of *prior*, the considered dependency Γ does *not* interfere with Ψ^A.

Beside the non-interference between *prior* and $\widehat{psec}^A_{prior} \cup \{\Psi^A\}$, the definition of admissible additional potential secrets also requires $\widehat{psec}_{prior} \cup \widehat{psec}^A_{prior} \cup \{\Psi^A\}$ to be a cleaned set. This follows from $\widehat{psec}_{prior} \cup \widehat{psec}^A_{prior}$ being a cleaned set and by constructing $\Psi^A = (\exists \boldsymbol{X})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ such that $\bar{t}_m \neq \tilde{t}_m$ holds for each $(\exists \boldsymbol{Z})\, R(\tilde{t}_1, \ldots, \tilde{t}_n) \in \widehat{psec}_{prior} \cup \widehat{psec}^A_{prior}$ with $\tilde{t}_m \in Dom$, as shown in the proof of Theorem 4.1 that interchangeability is well-defined.

Last but not least, requirement (iii) of Definition 5.10 – that the non-implication $\Gamma \not\models_{DB} \Psi^A$ should hold for each $\Gamma \in prior$ – is always satisfied: each $\Gamma \in prior$ in the form of a single premise tuple generating dependency is satisfied by an empty DB-Interpretation, but such an empty DB-Interpretation does *not* satisfy Ψ^A in the form of an existentially quantified atom. ♠

5.5 Inference-Proofness of the Adapted Approach

Now that the extended weakening algorithm – which is capable of handling an adversary's a priori knowledge in the form of single premise tuple generating dependencies – is formalized, its inference-proofness in the sense of Controlled Interaction Execution is also shown in a formal way. For that purpose, two preparing results are presented in the following as a preliminary step, and then the final proof relying on these preparing results is provided.

5.5.1 Preparing Results

For the first of these preparing results a partitioning $\mathcal{P}$ of an adversary's a priori knowledge *prior* is considered, which is constructed according to Definition 5.7 with respect to a considered confidentiality policy $psec_{prior}$, which is extended with respect to *prior*. Thereby, it is essentially shown that one can be sure that all dependencies of a partition $P_i \in \mathcal{P}$ interfere with the extended confidentiality policy $psec_{prior}$, if at least one dependency of this partition P_i is known to interfere with $psec_{prior}$. As a consequence of this insight, the corresponding cleaned confidentiality policy $\widehat{psec}_{prior}$ contains potential secrets $\Psi_{\text{prem}} \in \widehat{psec}_{prior}$ and $\Psi_{\text{concl}} \in \widehat{psec}_{prior}$ with $prem(\Gamma) \models_{DB} \Psi_{\text{prem}}$ and $concl(\Gamma) \models_{DB} \Psi_{\text{concl}}$ for each

dependency Γ of a partition $P_i \in \mathcal{P}$, if this partition P_i – and hence at least one of its dependencies – interferes with $\widehat{psec}_{prior}$.

Lemma 5.3: Completely Protected Partitions

Consider an adversary's a priori knowledge *prior* consisting of single premise tuple generating dependencies and a confidentiality policy $psec_{prior}$, which is extended with respect to *prior*. Moreover, suppose that $\widehat{psec}_{prior}$ is the cleaned policy of $psec_{prior}$ and further suppose that $\mathcal{P}$ is a partitioning of *prior* with respect to $\widehat{psec}_{prior}$ according to Definition 5.7.

If a partition $P_i \in \mathcal{P}$ interferes with $\widehat{psec}_{prior}$, then for each dependency $\Gamma \in P_i$ there are potential secrets $\Psi_{\text{prem}} \in \widehat{psec}_{prior}$ and $\Psi_{\text{concl}} \in \widehat{psec}_{prior}$ with $prem(\Gamma) \models_{DB} \Psi_{\text{prem}}$ and $concl(\Gamma) \models_{DB} \Psi_{\text{concl}}$.

Proof. Consider an arbitrary partition $P_i \in \mathcal{P}$ interfering with $\widehat{psec}_{prior}$. To establish the *proof by induction*, consider the sets $\mathcal{S}_j$ with $j \in \{0, \ldots, \ell\}$ and suppose that each such set $\mathcal{S}_j$ contains each dependency $\Gamma \in P_i$, for which the existence of potential secrets $\Psi_{\text{prem}} \in \widehat{psec}_{prior}$ and $\Psi_{\text{concl}} \in \widehat{psec}_{prior}$ with $prem(\Gamma) \models_{DB} \Psi_{\text{prem}}$ and $concl(\Gamma) \models_{DB} \Psi_{\text{concl}}$ is proved after the j-th iteration of induction.

To start with the *base clause*, consider the set $\mathcal{S}_0$: as the considered partition P_i is supposed to interfere with the cleaned policy $\widehat{psec}_{prior}$, there is at least one dependency $\Gamma \in P_i$ that interferes with $\widehat{psec}_{prior}$. Then, this dependency Γ also interferes with the non-cleaned policy $psec_{prior}$, which is a superset of the cleaned policy $\widehat{psec}_{prior}$. Additionally considering that the non-cleaned policy $psec_{prior}$ is extended with respect to *prior* according to Definition 5.5, this non-cleaned policy $psec_{prior}$ is guaranteed to contain both $prem(\Gamma)$ and $concl(\Gamma)$ as potential secrets. Hence, due to the properties of a cleaned set according to Definition 3.4, the cleaned confidentiality policy $\widehat{psec}_{prior}$ contains the (possibly weaker) potential secrets $\Psi_{\text{prem}} \in \widehat{psec}_{prior}$ and $\Psi_{\text{concl}} \in \widehat{psec}_{prior}$ with $prem(\Gamma) \models_{DB} \Psi_{\text{prem}}$ and $concl(\Gamma) \models_{DB} \Psi_{\text{concl}}$. As a direct consequence, each such dependency $\Gamma \in P_i$ that interferes with $\widehat{psec}_{prior}$ is in the set $\mathcal{S}_0$.

To prepare the induction step, the *induction hypothesis* is established as follows: for each $j \in \{0, \ldots, \ell-1\}$ the set $\mathcal{S}_j$ contains each $\Gamma \in P_i$, for which the existence of potential secrets $\Psi_{\text{prem}} \in \widehat{psec}_{prior}$ and $\Psi_{\text{concl}} \in \widehat{psec}_{prior}$ with $prem(\Gamma) \models_{DB} \Psi_{\text{prem}}$ and $concl(\Gamma) \models_{DB} \Psi_{\text{concl}}$ is proved after the j-th iteration of induction.

To now perform the *induction step*, consider an arbitrary $j \in \{0, \ldots, \ell - 1\}$. By construction of the partitioning $\mathcal{P}$ and because of $j < \ell$ there are dependencies

$\bar{\Gamma} \in \mathcal{S}_j \subseteq P_i$ and $\Gamma \in P_i$ with $\Gamma \notin \mathcal{S}_j$ such that

(a) there is a potential secret $\Psi \in \widehat{psec}_{prior}$ with both $concl(\bar{\Gamma})[\bar{\sigma}] \models_{DB} \Psi$ and $concl(\Gamma)[\sigma] \models_{DB} \Psi$ under arbitrary constant substitutions $\bar{\sigma}$ and σ,

(b) the implication $concl(\bar{\Gamma})[\bar{\sigma}] \models_{DB} prem(\Gamma)$ holds under an arbitrary constant substitution $\bar{\sigma}$ or

(c) the implication $concl(\Gamma)[\sigma] \models_{DB} prem(\bar{\Gamma})$ holds under an arbitrary constant substitution σ.

Case (a) relies on the assumption that there is a potential secret Ψ in the cleaned confidentiality policy $\widehat{psec}_{prior}$ with $concl(\Gamma)[\sigma] \models_{DB} \Psi$ under an arbitrary constant substitution σ. So, according to the definition of DB-Implication, this potential secret Ψ and $concl(\Gamma)$ share a common constant unifier and hence Γ interferes with the non-cleaned policy $psec_{prior}$, which also contains Ψ due to being a superset of the cleaned policy $\widehat{psec}_{prior}$.

Considering case (b), the induction hypothesis guarantees that there is a potential secret $\bar{\Psi}_{\text{concl}} \in \widehat{psec}_{prior}$ with $concl(\bar{\Gamma}) \models_{DB} \bar{\Psi}_{\text{concl}}$ because of $\bar{\Gamma} \in \mathcal{S}_j$. Hence, the non-cleaned confidentiality policy $psec_{prior}$ also contains $\bar{\Psi}_{\text{concl}}$ due to being a superset of the cleaned policy $\widehat{psec}_{prior}$. As this non-cleaned policy $psec_{prior}$ is moreover extended with respect to *prior* according to Definition 5.5, it is guaranteed to contain $concl(\bar{\Gamma})$ as a potential secret because of the interference established by $concl(\bar{\Gamma})$ and $\bar{\Psi}_{\text{concl}} \in psec_{prior}$, which share a common constant unifier due to the implication $concl(\bar{\Gamma}) \models_{DB} \bar{\Psi}_{\text{concl}}$. By further considering that the implication $concl(\bar{\Gamma})[\bar{\sigma}] \models_{DB} prem(\Gamma)$ is supposed to hold in this case under an arbitrary constant substitution $\bar{\sigma}$, there is a potential secret $\bar{\Psi} = concl(\bar{\Gamma})$ in the non-cleaned confidentiality policy $psec_{prior}$ for which the implication $\bar{\Psi}[\bar{\sigma}] \models_{DB} prem(\Gamma)$ holds under $\bar{\sigma}$ and as a consequence the dependency Γ interferes with the non-cleaned policy $psec_{prior}$ according to case (ii) of Definition 5.4.

Considering case (c), the induction hypothesis guarantees that there is a potential secret $\bar{\Psi}_{\text{prem}} \in \widehat{psec}_{prior}$ with $prem(\bar{\Gamma}) \models_{DB} \bar{\Psi}_{\text{prem}}$ because of $\bar{\Gamma} \in \mathcal{S}_j$. Hence, the non-cleaned confidentiality policy $psec_{prior}$ also contains $\bar{\Psi}_{\text{prem}}$ due to being a superset of the cleaned policy $\widehat{psec}_{prior}$. By further considering that the implication $concl(\Gamma)[\sigma] \models_{DB} prem(\bar{\Gamma})$ is supposed to hold in this case under the constant substitution σ, the implication $concl(\Gamma)[\sigma] \models_{DB} \bar{\Psi}_{\text{prem}}$ holds by transitivity and hence the definition of DB-Implication guarantees that $concl(\Gamma)$ and $\bar{\Psi}_{\text{prem}}$ share a common constant unifier. As a direct consequence, the dependency Γ interferes with the non-cleaned policy $psec_{prior}$ containing $\bar{\Psi}_{\text{prem}}$.

In each of the cases discussed above the dependency Γ interferes with the non-cleaned confidentiality policy $psec_{prior}$. Again considering that the non-cleaned

policy $psec_{prior}$ is extended with respect to $prior$ according to Definition 5.5, this non-cleaned policy $psec_{prior}$ is guaranteed to contain both $prem(\Gamma)$ and $concl(\Gamma)$ as potential secrets. Hence, due to the properties of a cleaned set according to Definition 3.4, the cleaned confidentiality policy $\widehat{psec}_{prior}$ contains the (possibly weaker) potential secrets $\Psi_{\text{prem}} \in \widehat{psec}_{prior}$ and $\Psi_{\text{concl}} \in \widehat{psec}_{prior}$ with $prem(\Gamma) \models_{DB} \Psi_{\text{prem}}$ and $concl(\Gamma) \models_{DB} \Psi_{\text{concl}}$. As a direct consequence, each such dependency Γ is in the set $\mathcal{S}_{j+1}$ in addition to all dependencies of $\mathcal{S}_j$. ♠

In Section 3.3 the inference-proofness of the generic weakening algorithm is verified by providing a generic method to construct confidentiality preserving alternative database instances on the basis of a so-called ground operator, which essentially generates database tuples inducing DB-Interpretations that satisfy exactly one sentence of a cleaned set of existentially quantified atoms. To verify the inference-proofness of the extended weakening algorithm, confidentiality preserving alternative database instances are similarly constructed on the basis of the following extension of this ground operator.

Definition 5.12: Extended Ground Operator

Consider an existentially quantified atom Φ constructed over a predicate symbol R of arity n and a finite set $\mathcal{S}$ of existentially quantified atoms all constructed over this predicate symbol R. Moreover, suppose that a non-empty but finite subset

$$Dom^* \subset Dom \setminus \{\, t_i \mid (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n) \in \mathcal{S} \;\text{ and }\; t_i \in Dom \;\text{ and }\; 1 \leq i \leq n \,\}$$

of constant symbols of Dom is given, i.e., a non-empty but finite subset of constant symbols which is *not* contained in the active domain of $\mathcal{S}$.

The (extended) *ground operator* $grnd(\Phi, \mathcal{S}, Dom^*)$ deterministically returns a constant combination $\boldsymbol{c} \in Dom^n$ such that the induced DB-Interpretation $\mathcal{I}_{\boldsymbol{c}}$ with $\mathcal{I}_{\boldsymbol{c}}(R) = \{\boldsymbol{c}\}$

(i) contains only constant symbols occurring in Φ and in Dom^*,

(ii) satisfies the sentence Φ, i.e., $\mathcal{I}_{\boldsymbol{c}} \models_M \Phi$, and

(iii) does *not* satisfy any sentence $\bar{\Phi}$ of the set

$$\mathcal{S}(\Phi) := \{\, \bar{\Phi} \in \mathcal{S} \mid \Phi \not\models_{DB} \bar{\Phi} \,\} \;,$$

i.e., $\mathcal{I}_{\boldsymbol{c}} \not\models_M \bar{\Phi}$ for each $\bar{\Phi} \in \mathcal{S}(\Phi)$.

Similar to the ground operator known from Definition 3.8, this extended ground operator also constructs database tuples, which induce DB-Interpretations satisfying a sentence Φ without (accidentally) satisfying certain other sentences, which are *not* implied by Φ and which do hence – according to the definition of DB-Implication – *not* necessarily need to be satisfied, if Φ is satisfied. Thereby, all fresh constant symbols of a database tuple constructed to satisfy a sentence Φ stem from a certain finite subset of constant symbols of the infinite domain Dom to guarantee that the active domains of alternative database instances constructed on the basis of the extended ground operator are limited to certain finite subsets of constant symbols of Dom.

As known from Section 3.3, it is now formally verified that this extended ground operator is well-defined to be able to rely on this operator within the main proof.

Lemma 5.4: Well-Defined Extended Ground Operator

Let Φ be an existentially quantified atom constructed over a predicate symbol R of arity n and let $\mathcal{S}$ be a finite set of existentially quantified atoms all constructed over this predicate symbol R. Moreover, let

$$Dom^* \subset Dom \setminus \{\, t_i \mid (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n) \in \mathcal{S} \text{ and } t_i \in Dom \text{ and } 1 \leq i \leq n \,\}$$

be a non-empty but finite subset of constant symbols of Dom. Then, the ground operator $grnd(\Phi, \mathcal{S}, Dom^*)$ is able to return a constant combination $\boldsymbol{c} \in Dom^n$ as required by Definition 5.12.

Proof. Consider an arbitrary existentially quantified atom $\Phi = (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n)$ and an arbitrary finite set $\mathcal{S}$ of existentially quantified atoms over predicate symbol R. To construct the constant combination $\boldsymbol{c} = (c_1, \ldots, c_n)$ to be returned by $grnd(\Phi, \mathcal{S}, Dom^*)$, set $c_i := t_i$ for each i with $t_i \in Dom$. Then, for each i with $t_i \in \boldsymbol{X}$ deterministically choose c_i from the non-empty subset Dom^* of constant symbols of Dom, which is *not* contained in the active domain of $\mathcal{S}$.

Obviously, this construction of $\boldsymbol{c}$ guarantees that each c_i is either a constant symbol occurring in Φ or a constant symbol of Dom^* and that the induced DB-Interpretation $\mathcal{I}_{\boldsymbol{c}}$ with $\mathcal{I}_{\boldsymbol{c}}(R) = \{\boldsymbol{c}\}$ satisfies the existentially quantified atom Φ, i.e., $\mathcal{I}_{\boldsymbol{c}} \models_M \Phi$. To moreover assure that this DB-Interpretation $\mathcal{I}_{\boldsymbol{c}}$ does *not* satisfy any existentially quantified atom of the set $\mathcal{S}(\Phi)$, assume that there is such a sentence $\bar{\Phi} = (\exists \boldsymbol{Y})\, R(\bar{t}_1, \ldots, \bar{t}_n)$ in $\mathcal{S}(\Phi)$ with $\mathcal{I}_{\boldsymbol{c}} \models_M \bar{\Phi}$. Then, for each i with $t_i \in Dom$ either $\bar{t}_i = c_i = t_i$ or $\bar{t}_i \in \boldsymbol{Y}$ holds and for each i with $t_i \in \boldsymbol{X}$ the term

$\bar{t}_i$ must be a variable of $\boldsymbol{Y}$ because of choosing c_i from the subset Dom^* of constant symbols *not* containing any constant symbol occurring in a sentence of $\mathcal{S}(\Phi)$. Hence, by applying Lemma 2.1, the implication $\Phi \models_{DB} \bar{\Phi}$ holds in contradiction to the construction of the set $\mathcal{S}(\Phi) := \{\, \bar{\Phi} \in \mathcal{S} \mid \Phi \not\models_{DB} \bar{\Phi} \,\}$. ♠

5.5.2 Establishing the Main Result

Now, the inference-proofness of the extended weakening algorithm is formally verified. Similar to the proof developed for the generic weakening algorithm in Section 3.3, a generic method for the construction of alternative database instances is provided, which are both indistinguishable from a considered original database instance from an adversary's point of view and do *not* satisfy a particular potential secret of a considered confidentiality policy. Such an alternative instance hence serves as a witness that the considered potential secret does *not* necessarily need to be satisfied from an adversary's point of view.

Theorem 5.2: Inference-Proofness under A Priori Knowledge

Let r be a complete database instance over a database schema $\langle R|\mathcal{A}_R|SC_R\rangle$, let *psec* be a confidentiality policy of existentially quantified atoms and suppose that a notion of admissible indistinguishabilities is given. Moreover, assume that an adversary's a priori knowledge *prior* (with $SC_R \subseteq prior$) consisting of single premise tuple generating dependencies is given such that $\mathcal{I}_r \models_M prior$ holds and $prior \not\models_{DB} \Psi$ is valid for each $\Psi \in psec$.

Algorithm 5.1 then creates a weakened view $weak(r, psec, prior)$ on the given database instance r, which is inference-proof in the sense that for each potential secret $\Psi \in psec$ the existence of a complete alternative instance r^{Ψ} over schema $\langle R|\mathcal{A}_R|SC_R\rangle$ is guaranteed. This alternative instance r^{Ψ}

(i) obeys the potential secret Ψ, i.e., $\mathcal{I}_{r^{\Psi}} \not\models_M \Psi$,

(ii) satisfies the adversary's a priori knowledge, i.e., $\mathcal{I}_{r^{\Psi}} \models_M prior$, and

(iii) the weakened view $weak(r^{\Psi}, psec, prior)$ on the alternative instance r^{Ψ} is indistinguishable from the weakened view $weak(r, psec, prior)$ on the original instance r, i.e., $weak(r^{\Psi}, psec, prior) = weak(r, psec, prior)$.

Proof. Consider an arbitrary potential secret $\tilde{\Psi} \in psec$ to be protected and suppose that Stage 1 of Algorithm 5.1 – which does *not* depend on the original instance r at all – finished successfully with the generation of a partly extended matching M^+

consisting of pairwise disjoint clusters, each of which is of size 2. By construction of the extended and cleaned confidentiality policy $\widehat{psec}_{prior}$ there is a (weakest) potential secret $\hat{\Psi} \in \widehat{psec}_{prior}$ such that the implication $\tilde{\Psi} \models_{DB} \hat{\Psi}$ holds.

If the potential secret $\hat{\Psi}$ is *not* satisfied by the original instance r, i.e., $\mathcal{I}_r \not\models_M \hat{\Psi}$, the complete alternative instance $r^{\hat{\Psi}}$ protecting $\hat{\Psi}$ is simply r itself, i.e., $r^{\hat{\Psi}} := r$. This construction of $r^{\hat{\Psi}}$ directly implies $\mathcal{I}_{r^{\hat{\Psi}}} \not\models_M \hat{\Psi}$ and consequently the constructed alternative instance $r^{\hat{\Psi}}$ obeys $\hat{\Psi}$, i.e., $\mathcal{I}_{r^{\hat{\Psi}}} \not\models_M \hat{\Psi}$. Hence, by $\tilde{\Psi} \models_{DB} \hat{\Psi}$ and by applying Lemma 3.1, the alternative instance $r^{\hat{\Psi}}$ also obeys $\tilde{\Psi}$, i.e., $\mathcal{I}_{r^{\hat{\Psi}}} \not\models_M \tilde{\Psi}$. Moreover, $r^{\hat{\Psi}}$ satisfies the adversary's a priori knowledge *prior*, i.e., $\mathcal{I}_{r^{\hat{\Psi}}} \models_M prior$, because of both $r^{\hat{\Psi}} = r$ and $\mathcal{I}_r \models_M prior$ and the property of indistinguishability, i.e., $weak(r^{\hat{\Psi}}, psec, prior) = weak(r, psec, prior)$, is a direct consequence of $r^{\hat{\Psi}} = r$.

If the potential secret $\hat{\Psi}$ is satisfied by the original instance r, i.e., $\mathcal{I}_r \models_M \hat{\Psi}$, a complete alternative instance $r^{\hat{\Psi}}$ protecting $\hat{\Psi}$ can be constructed with the help of the (extended) ground operator introduced in Definition 5.12. But as a preprocessing step, it is to check first, whether there is a partition $P_i \in \mathcal{P}$ containing a dependency $\Gamma \in P_i$, for which the implication $concl(\Gamma)[\sigma] \models_{DB} \hat{\Psi}$ holds under an arbitrary constant substitution σ. If this is the case, set $m := i$. Otherwise, this variable m is supposed to be undefined. Next, the finite set $\mathcal{S}_{\hat{\Psi}}$ containing

- the (possibly additional) potential secrets of each cluster $C \in M^+$,
- the (non-additional) refused potential secrets of $\widehat{psec}_{prior} \setminus \bigcup_{C \in M^+} C$ not occurring in any cluster of M^+,
- the existentially quantified premise $prem(\Gamma)$ and the existentially quantified conclusion $concl(\Gamma)$ of each dependency Γ being in the partition P_m, if the variable m is defined, and
- the existentially quantified premise $prem(\Gamma)$ of each dependency Γ being in a partition $P_j \in \mathcal{P}$ *not* interfering with $\widehat{psec}_{prior}$

is constructed as the set of all existentially quantified atoms *not* to be satisfied (if possible) by the ground operator.

To further prepare the construction of the alternative instance $r^{\hat{\Psi}}$ with the help of the (extended) ground operator, consider an arbitrary non-empty but finite subset

$$Dom^* \subset Dom \setminus \{\, t_i \mid (\exists \boldsymbol{X})\, R(t_1, \ldots, t_n) \in \mathcal{S}_{\hat{\Psi}} \text{ and } t_i \in Dom \text{ and } 1 \leq i \leq n \,\}$$

of constant symbols of Dom, i.e., a non-empty but finite subset of "fresh" constant symbols, which is *not* contained in the active domain of the set $\mathcal{S}_{\hat{\Psi}}$ constructed

above. This construction is always possible as Dom is supposed to be an infinite set of constant symbols while the set $\mathcal{S}_{\hat{\Psi}}$ only contains a finite number of existentially quantified atoms each of which is of finite arity.

Finally, the complete alternative instance $r^{\hat{\Psi}}$ protecting the potential secret $\hat{\Psi}$ can be constructed by adding

(i) the tuple $\boldsymbol{c}$ for each ground atom $R(\boldsymbol{c}) \in \mathit{weak}(r, \mathit{psec}, \mathit{prior})^+$,

(ii) the tuple $\mathit{grnd}(\mathit{concl}(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*)$ for each dependency $\Gamma \in P_i$ of each partition $P_i \in \mathcal{P}$ with[5] $i \neq m$, which moreover interferes with $\widehat{\mathit{psec}}_{\mathit{prior}}$, under each possible constant substitution σ substituting the universally quantified variables of Γ with constant symbols of

- the (possibly additional) potential secrets of the clusters of M^+,
- the dependencies of the adversary's a priori knowledge prior, and
- the fresh constant symbols of the set Dom^*, and

(iii) the tuple $\mathit{grnd}(\Psi, \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*)$ for an arbitrary (possibly additional) potential secret Ψ of each cluster $C \in M_r^+$ such that

- $\Psi \neq \hat{\Psi}$ and
- $\Psi \not\models_{DB} \mathit{prem}(\Gamma)$ and $\Psi \not\models_{DB} \mathit{concl}(\Gamma)$ for each $\Gamma \in P_m$, if the variable m is defined,

to this initially empty database instance $r^{\hat{\Psi}}$. Note that the sequence in which these tuples are added to $r^{\hat{\Psi}}$ is *not* of importance, as the ground operator only depends on inputs which remain unchanged during the construction of $r^{\hat{\Psi}}$. In particular, the clustering stage of Algorithm 5.1 does *not* depend at all on the database instance considered and the considered original instance r – and hence also the set M_r^+ of all clusters, whose corresponding disjunctions are satisfied by r – is *not* affected by the construction of the alternative instance $r^{\hat{\Psi}}$.

To show: $\mathcal{I}_{r^{\hat{\Psi}}} \not\models_M \tilde{\Psi}$

By the above given construction of the alternative instance $r^{\hat{\Psi}}$, the induced DB-Interpretation $\mathcal{I}_{r^{\hat{\Psi}}}$ does *not* satisfy the potential secret $\hat{\Psi}$, i.e., there is *no* single tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$ inducing a DB-Interpretation satisfying $\hat{\Psi}$. First of all, each tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$ stemming from a ground atom $R(\boldsymbol{c}) \in \mathit{weak}(r, \mathit{psec}, \mathit{prior})^+$ according to construction rule (i) does *not* induce such a DB-Interpretation: considering the

[5] If m is undefined, $i \neq m$ is supposed to hold for each $P_i \in \mathcal{P}$.

assumption $\mathcal{I}_r \models_M \hat{\Psi}$, the potential secret $\hat{\Psi}$ is either in a cluster $C \in M_r^+$ or in the set of all refused potential secrets of $\widehat{psec}_{prior} \setminus \bigcup_{C \in M^+} C$ not occurring in any cluster of M^+ and in both of these cases the construction of $weak(r, psec, prior)^+$ guarantees that the non-implication $R(\boldsymbol{c}) \not\models_{DB} \hat{\Psi}$ holds according to Definition 5.11.

Moreover, each tuple $grnd(concl(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, Dom^*) \in r^{\hat{\Psi}}$, which is constructed for a dependency $\Gamma \in P_i$ of a partition $P_i \in \mathcal{P}$ with $i \neq m$ according to construction rule (ii), does *not* induce a DB-Interpretation satisfying the potential secret $\hat{\Psi}$. If the variable m is undefined, there is *no* partition $P_i \in \mathcal{P}$ containing a dependency $\Gamma \in P_i$ with $concl(\Gamma)[\sigma] \models_{DB} \hat{\Psi}$ under an arbitrary constant substitution σ. Hence, as the potential secret $\hat{\Psi}$ of $\widehat{psec}_{prior}$ is further contained in the set $\mathcal{S}_{\hat{\Psi}}$ of all existentially quantified atoms *not* to be satisfied (if possible) by the ground operator because of either $\hat{\Psi} \in \bigcup_{C \in M^+} C$ or $\hat{\Psi} \in (\widehat{psec}_{prior} \setminus \bigcup_{C \in M^+} C)$, each tuple $grnd(concl(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, Dom^*) \in r^{\hat{\Psi}}$ does *not* induce a DB-Interpretation satisfying $\hat{\Psi}$ by construction of the ground operator according to Lemma 5.4.

If m is defined and there hence is a partition $P_m \in \mathcal{P}$ containing a dependency $\bar{\Gamma} \in P_m$ with $concl(\bar{\Gamma})[\bar{\sigma}] \models_{DB} \hat{\Psi}$ under an arbitrary constant substitution $\bar{\sigma}$, for *none* of the dependencies $\Gamma \in P_m$ a tuple $grnd(concl(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, Dom^*)$ is added to $r^{\hat{\Psi}}$ according to construction rule (ii). Furthermore, each other partition $P_i \in \mathcal{P}$ with $i \neq m$ does *not* contain such a dependency Γ with $concl(\Gamma)[\sigma] \models_{DB} \hat{\Psi}$ under an arbitrary constant substitution σ: otherwise, $i \neq m$ can *not* hold as all partitions of *prior* are supposed to be pairwise disjoint and as the partitioning of *prior* would moreover require that the (different) dependencies Γ of P_i and $\bar{\Gamma}$ of P_m are in the same partition of $\mathcal{P}$ because of $concl(\Gamma)[\sigma]$ and $concl(\bar{\Gamma})[\bar{\sigma}]$ both implying *the same* potential secret $\hat{\Psi}$ of $\widehat{psec}_{prior}$. Hence, again considering that $\hat{\Psi}$ is in the set $\mathcal{S}_{\hat{\Psi}}$, each tuple $grnd(concl(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, Dom^*) \in r^{\hat{\Psi}}$ does *not* induce a DB-Interpretation satisfying $\hat{\Psi}$ according to Lemma 5.4.

Each tuple $grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*) \in r^{\hat{\Psi}}$, which is constructed for a (possibly additional) potential secret Ψ of a cluster $C \in M_r^+$ according to construction rule (iii), does *not* induce a DB-Interpretation satisfying the potential secret $\hat{\Psi}$, either. A tuple $grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*)$ is only added to $r^{\hat{\Psi}}$ for a potential secret Ψ with $\Psi \neq \hat{\Psi}$. If Ψ is a non-additional potential secret of the cleaned policy $\widehat{psec}_{prior}$, which also contains $\hat{\Psi}$, the non-implication $\Psi \not\models_{DB} \hat{\Psi}$ is guaranteed by the properties of a cleaned set. If Ψ is an additional potential secret, the non-implication $\Psi \not\models_{DB} \hat{\Psi}$ is guaranteed by the requirement that the set of all additional potential secrets occurring in a partly extended matching must be admissible in the sense of Definition 5.10, according to which the union of all additional potential secrets and all non-additional potential secrets must again be a cleaned set. Hence, because of $\hat{\Psi}$ being in the set $\mathcal{S}_{\hat{\Psi}}$ of all existentially quantified atoms *not* to be satisfied

(if possible) by the ground operator, each tuple $grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*)$ added to $r^{\hat{\Psi}}$ does *not* induce a DB-Interpretation satisfying $\hat{\Psi}$ by construction of the ground operator according to Lemma 5.4.

Summing up the above presented results, the proof of $\mathcal{I}_{r^{\hat{\Psi}}} \not\models_M \hat{\Psi}$ is achieved. By furthermore considering $\tilde{\Psi} \models_{DB} \hat{\Psi}$ and Lemma 3.1, the constructed alternative instance $r^{\hat{\Psi}}$ also obeys the potential secret $\tilde{\Psi}$ to be protected, i.e., $\mathcal{I}_{r^{\hat{\Psi}}} \not\models_M \tilde{\Psi}$.

To show: $weak(r^{\hat{\Psi}}, psec, prior) = weak(r, psec, prior)$

Next, it is to show that the weakened view $weak(r^{\hat{\Psi}}, psec, prior)$ on the constructed complete alternative instance $r^{\hat{\Psi}}$ is indistinguishable from the weakened view $weak(r, psec, prior)$ on the original instance r from an adversary's point of view. As a first step, $weak(r^{\hat{\Psi}}, psec, prior)^{\vee} = weak(r, psec, prior)^{\vee}$ is proved now.

Subcase: $weak(r^{\hat{\Psi}}, psec, prior)^{\vee} = weak(r, psec, prior)^{\vee}$

For this subcase consider an arbitrary (extended) cluster $C \in M^+$ and suppose that the original instance r satisfies the corresponding disjunction $\bigvee_{\Psi \in C} \Psi$, i.e., $\mathcal{I}_r \models_M \bigvee_{\Psi \in C} \Psi$. By this assumption, the cluster C is in the set M_r^+ of all clusters satisfied by r and by rule (iii) of the construction of the alternative instance $r^{\hat{\Psi}}$ a tuple $grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*)$ is added to $r^{\hat{\Psi}}$ for an arbitrary (possibly additional) potential secret Ψ of the cluster C with $\Psi \neq \hat{\Psi}$ and, if the variable m is defined, with further $\Psi \not\models_{DB} prem(\Gamma)$ and $\Psi \not\models_{DB} concl(\Gamma)$ for each $\Gamma \in P_m$. Hence, according to Lemma 5.4, the alternative instance $r^{\hat{\Psi}}$ also satisfies the disjunction $\bigvee_{\Psi \in C} \Psi$, i.e., $\mathcal{I}_{r^{\hat{\Psi}}} \models_M \bigvee_{\Psi \in C} \Psi$, provided that the existence of such a (possibly additional) potential secret $\Psi \in C$ as required above is guaranteed.

- If the variable m is undefined, there is such a potential secret $\Psi \in C$ with $\Psi \neq \hat{\Psi}$ due to the properties of a partly extended matching guaranteeing that C contains 2 semantically different potential secrets.
- If the variable m is defined and the cluster C does *not* contain an additional potential secret, the construction of the indistinguishability graph $G = (V, E)$ ensures that for the edge $C = \{\Psi, \Psi'\} \in E$ of this graph – which corresponds to the cluster C – there is *no* single partition $P_i \in \mathcal{P}$ containing dependencies $\Gamma_1 \in P_i$ and $\Gamma_2 \in P_i$ (with possibly $\Gamma_1 = \Gamma_2$) such that both implications $\Psi \models_{DB} prem(\Gamma_1)$ and $\Psi' \models_{DB} prem(\Gamma_2)$ hold. As a consequence, at least one of the potential secrets of the cluster C does *not* imply the existentially quantified premise of any dependency $\Gamma \in P_m$.

Moreover, for both potential secrets Ψ and Ψ' of this cluster C the non-implications $\Psi \not\models_{DB} concl(\Gamma)$ and $\Psi' \not\models_{DB} concl(\Gamma)$ hold for each dependency $\Gamma \in P_m$. Otherwise, if there is such a dependency $\Gamma \in P_m$ with $\bar{\Psi} \models_{DB} concl(\Gamma)$ for a potential secret $\bar{\Psi} \in C$, this dependency Γ interferes with $\widehat{psec}_{prior}$ due to $\bar{\Psi}$ and $concl(\Gamma)$ sharing a common constant unifier according to the definition of DB-Implication. Hence, the extended and cleaned policy $\widehat{psec}_{prior}$ also contains a potential secret Ψ_{concl} with $concl(\Gamma) \models_{DB} \Psi_{\text{concl}}$ and the implication $\bar{\Psi} \models_{DB} \Psi_{\text{concl}}$ holds by transitivity. Further, the sentences $\bar{\Psi}$ and Ψ_{concl} can *not* be semantically equivalent because of $concl(\Gamma) \models_{DB} \Psi_{\text{concl}}$ and because of $concl(\Gamma) \not\models_{DB} \bar{\Psi}$ due to $\bar{\Psi} \in C$ and considering that all non-additional potential secrets of clusters of M^+ stem from the subset $\widehat{psec}_{prior} \setminus concl(\widehat{psec}_{prior}, prior)$ of potential secrets, which serves as the set of vertices of an indistinguishability graph (cf. Definition 5.8 and Definition 5.6). As an immediate consequence, the properties of a cleaned set guarantee that the (stronger) potential secret $\bar{\Psi}$ can – in contradiction to the assumption – *not* be in the cleaned set $\widehat{psec}_{prior}$ as long as the (weaker) potential secret Ψ_{concl} also is in $\widehat{psec}_{prior}$.

So, there is at least one potential secret $\Psi \in C$ with both $\Psi \not\models_{DB} prem(\Gamma)$ and $\Psi \not\models_{DB} concl(\Gamma)$ for each $\Gamma \in P_m$. As m is supposed to be defined, there is a dependency $\Gamma \in P_m$ with $concl(\Gamma)[\sigma] \models_{DB} \hat{\Psi}$ under an arbitrary constant substitution σ. Consequently, the potential secret $\hat{\Psi}$ is in the set $concl(\widehat{psec}_{prior}, prior)$. But, again considering that all non-additional potential secrets of clusters of M^+ stem from $\widehat{psec}_{prior} \setminus concl(\widehat{psec}_{prior}, prior)$, the potential secret $\hat{\Psi}$ can *not* be in the cluster C consisting only of non-additional potential secrets. Hence, $\Psi \neq \hat{\Psi}$ is an immediate consequence.

- If the variable m is defined and the (extended) cluster C contains an additional potential secret Ψ^A, the construction of admissible additional potential secrets according to Definition 5.10 guarantees that each dependency Γ of the partition P_m does *not* interfere with Ψ^A. Hence, both non-implications $\Psi^A \not\models_{DB} prem(\Gamma)$ and $\Psi^A \not\models_{DB} concl(\Gamma)$ hold for each $\Gamma \in P_m$ by the properties of this non-interference. The construction of admissible additional potential secrets moreover guarantees the inequality $\Psi^A \neq \hat{\Psi}$, as the union of all additional potential secrets (such as Ψ^A) and all non-additional potential secrets (such as $\hat{\Psi}$) must be a cleaned set.

For each (extended) cluster $C \in M^+$, whose corresponding disjunction $\bigvee_{\Psi \in C} \Psi$ is *not* satisfied by the original instance r, this corresponding disjunction $\bigvee_{\Psi \in C} \Psi$ is *not* satisfied by the constructed alternative instance $r^{\hat{\Psi}}$, either. Hence, there is no single tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$ inducing a DB-Interpretation satisfying a (possibly additional)

potential secret Ψ of the cluster C. First of all, each tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$ stemming from a ground atom $R(\boldsymbol{c}) \in \mathit{weak}(r, \mathit{psec}, \mathit{prior})^+$ according to construction rule (i) does *not* induce such a DB-Interpretation: the assumption $\mathcal{I}_r \not\models_M \bigvee_{\Psi \in C} \Psi$ guarantees that each tuple of the original instance r does *not* induce a DB-Interpretation satisfying a potential secret Ψ of the cluster C and for each ground atom $R(\boldsymbol{c}) \in \mathit{weak}(r, \mathit{psec}, \mathit{prior})^+$ the tuple $\boldsymbol{c}$ stems from this original instance r.

Moreover, each tuple $\mathit{grnd}(\mathit{concl}(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*) \in r^{\hat{\Psi}}$, which is constructed for a dependency $\Gamma \in P_i$ of a partition $P_i \in \mathcal{P}$ according to construction rule (ii), does *not* induce a DB-Interpretation satisfying the disjunction $\bigvee_{\Psi \in C} \Psi$. For each non-additional potential secret Ψ of the considered cluster C, the non-implication $\mathit{concl}(\Gamma)[\sigma] \not\models_{DB} \Psi$ holds under each constant substitution σ – otherwise Ψ would be in the set $\mathit{concl}(\widehat{\mathit{psec}}_{\mathit{prior}}, \mathit{prior})$ and could hence *not* be in a cluster of M^+. For each additional potential secret Ψ^A of the cluster C, the validity of the non-implication $\mathit{concl}(\Gamma)[\sigma] \not\models_{DB} \Psi^A$ is guaranteed under each constant substitution σ by the construction of admissible additional potential secrets ensuring that each dependency $\Gamma \in \mathit{prior}$ does *not* interfere with an additional potential secret. As hence the non-implication $\mathit{concl}(\Gamma)[\sigma] \not\models_{DB} \Psi$ holds for each (possibly additional) potential secret $\Psi \in C$ under each constant substitution σ and as each (possibly additional) potential secret of a cluster of M^+ is contained in the set $\mathcal{S}_{\hat{\Psi}}$ of all existentially quantified atoms *not* to be satisfied (if possible) by the ground operator, Lemma 5.4 guarantees that such a tuple $\mathit{grnd}(\mathit{concl}(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*)$ does *not* induce a DB-Interpretation satisfying the disjunction $\bigvee_{\Psi \in C} \Psi$.

There is also *no* tuple $\mathit{grnd}(\bar{\Psi}, \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*) \in r^{\hat{\Psi}}$ constructed for a (possibly additional) potential secret $\bar{\Psi}$ of a cluster $\bar{C} \in M_r^+$ according to construction rule (iii), which induces a DB-Interpretation satisfying the disjunction $\bigvee_{\Psi \in C} \Psi$. The assumption $\mathcal{I}_r \not\models_M \bigvee_{\Psi \in C} \Psi$ guarantees that the considered cluster C is *not* contained in the set M_r^+ of those clusters, whose corresponding disjunctions are satisfied by the original instance r. As moreover each potential secret $\bar{\Psi}$ of a cluster $\bar{C} \in M_r^+$ is *not* contained in the (different) cluster $C \notin M_r^+$ due to the disjoint clustering, there is *no* tuple $\mathit{grnd}(\Psi, \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*)$ added to $r^{\hat{\Psi}}$ for a potential secret $\Psi \in C$. For all (possibly additional) potential secrets $\bar{\Psi}$ of a cluster $\bar{C} \in M_r^+$, for which a tuple $\mathit{grnd}(\bar{\Psi}, \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*)$ is added to $r^{\hat{\Psi}}$, the non-implication $\bar{\Psi} \not\models_{DB} \Psi$ is ensured for each (possibly additional) potential secret $\Psi \in C$ by the properties of the cleaned set $\widehat{\mathit{psec}}_{\mathit{prior}}$ and by the construction of admissible additional potential secrets according to Definition 5.10. As each (possibly additional) potential secret of a cluster of M^+ is in the set $\mathcal{S}_{\hat{\Psi}}$ of all existentially quantified atoms *not* to be satisfied (if possible) by the ground operator, Lemma 5.4 again guarantees that a tuple $\mathit{grnd}(\bar{\Psi}, \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*)$ constructed for a potential secret $\bar{\Psi}$ of a cluster $\bar{C} \in M_r^+$ does *not* induce a DB-Interpretation satisfying the disjunction $\bigvee_{\Psi \in C} \Psi$.

Summing up the results proved above, a disjunction $\bigvee_{\Psi \in C} \Psi$ corresponding to a cluster $C \in M^+$ is satisfied by the alternative instance $r^{\hat{\Psi}}$ *if and only if* it is satisfied by the original instance r. So, the validity of $M^+_{r^{\hat{\Psi}}} = M^+_r$ and hence also $\mathit{weak}(r^{\hat{\Psi}}, \mathit{psec}, \mathit{prior})^\vee = \mathit{weak}(r, \mathit{psec}, \mathit{prior})^\vee$ is a direct consequence.

Subcase: $\mathit{weak}(r^{\hat{\Psi}}, \mathit{psec}, \mathit{prior})^+ = \mathit{weak}(r, \mathit{psec}, \mathit{prior})^+$

To next start with the subcase of $\mathit{weak}(r^{\hat{\Psi}}, \mathit{psec}, \mathit{prior})^+ = \mathit{weak}(r, \mathit{psec}, \mathit{prior})^+$, consider an arbitrary ground atom $R(\boldsymbol{c}) \in \mathit{weak}(r, \mathit{psec}, \mathit{prior})^+$. The construction of $\mathit{weak}(r, \mathit{psec}, \mathit{prior})^+$ guarantees the non-implication $R(\boldsymbol{c}) \not\models_{DB} \Psi$ for each (possibly additional) potential secret Ψ of a cluster of M^+_r and for each (non-additional) potential secret Ψ of $\widehat{\mathit{psec}}_{\mathit{prior}} \setminus \bigcup_{C \in M^+} C$ not occurring in any cluster of M^+. According to rule (i) of the construction of $r^{\hat{\Psi}}$, the tuple $\boldsymbol{c}$ corresponding to the ground atom $R(\boldsymbol{c})$ is contained in the alternative instance $r^{\hat{\Psi}}$. Hence, by construction of a weakened view according to Definition 5.11, the ground atom $R(\boldsymbol{c})$ is in $\mathit{weak}(r^{\hat{\Psi}}, \mathit{psec}, \mathit{prior})^+$, as the non-implication $R(\boldsymbol{c}) \not\models_{DB} \Psi$ also holds for each (possibly additional) potential secret of a cluster of $M^+_{r^{\hat{\Psi}}}$ because of $M^+_{r^{\hat{\Psi}}} = M^+_r$ (as proved above) and for each potential secret Ψ of the set $\widehat{\mathit{psec}}_{\mathit{prior}} \setminus \bigcup_{C \in M^+} C$, whose construction does *not* depend at all on the database instance considered.

For each tuple $\mathit{grnd}(\mathit{concl}(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*) \in r^{\hat{\Psi}}$, which is constructed according to construction rule (ii) for a dependency $\Gamma \in P_i$ of a partition $P_i \in \mathcal{P}$ interfering with the extended and cleaned confidentiality policy $\widehat{\mathit{psec}}_{\mathit{prior}}$, the corresponding ground atom $R(\mathit{grnd}(\mathit{concl}(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*))$ is *not* contained in $\mathit{weak}(r^{\hat{\Psi}}, \mathit{psec}, \mathit{prior})^+$. As the partition P_i interferes with the policy $\widehat{\mathit{psec}}_{\mathit{prior}}$, Lemma 5.3 guarantees the existence of a potential secret $\Psi_{\text{concl}} \in \widehat{\mathit{psec}}_{\mathit{prior}}$ with $\mathit{concl}(\Gamma) \models_{DB} \Psi_{\text{concl}}$. Hence, there is also a constant substitution σ under which the implication $\mathit{concl}(\Gamma)[\sigma] \models_{DB} \Psi_{\text{concl}}$ holds. As a consequence, the potential secret Ψ_{concl} is in the set $\mathit{concl}(\widehat{\mathit{psec}}_{\mathit{prior}}, \mathit{prior})$ of potential secrets to be refused (cf. Definition 5.6) and hence also in the set $\widehat{\mathit{psec}}_{\mathit{prior}} \setminus \bigcup_{C \in M^+} C$ of policy elements not occurring in any cluster of M^+. As moreover the implication $R(\mathit{grnd}(\mathit{concl}(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*)) \models_{DB} \Psi_{\text{concl}}$ holds because of both

- $R(\mathit{grnd}(\mathit{concl}(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*)) \models_{DB} \mathit{concl}(\Gamma)$ and
- $\mathit{concl}(\Gamma) \models_{DB} \Psi_{\text{concl}}$,

the ground atom $R(\mathit{grnd}(\mathit{concl}(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*))$ corresponding to the database tuple $\mathit{grnd}(\mathit{concl}(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*) \in r^{\hat{\Psi}}$ is *not* qualified for being in the positive knowledge $\mathit{weak}(r^{\hat{\Psi}}, \mathit{psec}, \mathit{prior})^+$ according to Definition 5.11.

Similarly, for each tuple $grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*) \in r^{\hat{\Psi}}$, which is constructed for a (possibly additional) potential secret Ψ of a cluster $C \in M_r^+$ according to construction rule (iii), the corresponding ground atom $R(grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*))$ is *not* contained in $weak(r^{\hat{\Psi}}, psec, prior)^+$, either. Because of $M^+_{r^{\hat{\Psi}}} = M_r^+$ (as proved above), the considered cluster C of M_r^+ is also in $M^+_{r^{\hat{\Psi}}}$. As there hence is the potential secret Ψ of the cluster C of $M^+_{r^{\hat{\Psi}}}$, for which the implication $R(grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*)) \models_{DB} \Psi$ holds, the considered ground atom $R(grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*))$ is *not* qualified for being in $weak(r^{\hat{\Psi}}, psec, prior)^+$ according to Definition 5.11. This finally results in $weak(r^{\hat{\Psi}}, psec, prior)^+ = weak(r, psec, prior)^+$.

Completing: $weak(r^{\hat{\Psi}}, psec, prior) = weak(r, psec, prior)$

According to Definition 5.11, the construction of the refused knowledge of a weakened view is only based on the subset $\widehat{psec}_{prior} \setminus \bigcup_{C \in M^+} C$ of those policy elements of $\widehat{psec}_{prior}$ *not* occurring in any cluster of M^+. As the construction of the extended and cleaned confidentiality policy $\widehat{psec}_{prior}$ and the construction of the partly extended matching M^+ do both *not* depend on the considered database instance at all, the validity of $weak(r^{\hat{\Psi}}, psec, prior)^? = weak(r, psec, prior)^?$ is guaranteed.

Reconsidering the construction of the negative knowledge of a weakened view according to Definition 5.11 – which consists of a sequence of negated disjunctions and a (partial) completeness sentence – the validity of

- $weak(r^{\hat{\Psi}}, psec, prior)^+ = weak(r, psec, prior)^+$,
- $weak(r^{\hat{\Psi}}, psec, prior)^\vee = weak(r, psec, prior)^\vee$ and
- $weak(r^{\hat{\Psi}}, psec, prior)^? = weak(r, psec, prior)^?$ as well as
- $M^+ \setminus M^+_{r^{\hat{\Psi}}} = M^+ \setminus M_r^+$ (due to $M^+_{r^{\hat{\Psi}}} = M_r^+$)

leads to $weak(r^{\hat{\Psi}}, psec, prior)^- = weak(r, psec, prior)^-$. Hence, indistinguishability is achieved because of $weak(r^{\hat{\Psi}}, psec, prior) = weak(r, psec, prior)$, provided that the sentences of both of these sequences are arranged in the same order.

To show: $\mathcal{I}_{r^{\hat{\Psi}}} \models_M prior$

Finally, it is to show that the constructed alternative instance $r^{\hat{\Psi}}$ satisfies the adversary's a priori knowledge *prior*, i.e., $\mathcal{I}_{r^{\hat{\Psi}}} \models_M prior$. As a first step towards this goal, consider an arbitrary dependency $\Gamma \in prior$ and suppose that Γ is in a partition $P_j \in \mathcal{P}$, which does *not* interfere with the confidentiality policy $\widehat{psec}_{prior}$.

Subcase: Γ of a partition $P_j \in \mathcal{P}$, which does *not* interfere with $\widehat{psec}_{prior}$

If the alternative instance $r^{\hat{\Psi}}$ does *not* satisfy $prem(\Gamma)[\sigma]$ under *any* constant substitution σ, i.e., $\mathcal{I}_{r^{\hat{\Psi}}} \not\models_M prem(\Gamma)[\sigma]$, the validity of $\mathcal{I}_{r^{\hat{\Psi}}} \models_M \Gamma$ is a direct consequence. If the alternative instance $r^{\hat{\Psi}}$ instead satisfies $prem(\Gamma)[\sigma]$ under a constant substitution σ, i.e., $\mathcal{I}_{r^{\hat{\Psi}}} \models_M prem(\Gamma)[\sigma]$, it needs to be shown that $\mathcal{I}_{r^{\hat{\Psi}}} \models_M concl(\Gamma)[\sigma]$ holds under this constant substitution σ, too.

The assumption $\mathcal{I}_{r^{\hat{\Psi}}} \models_M prem(\Gamma)[\sigma]$ guarantees that there is a tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$ inducing a DB-Interpretation satisfying $prem(\Gamma)[\sigma]$. This tuple $\boldsymbol{c}$ is *not* equal to a tuple $grnd(concl(\bar{\Gamma})[\bar{\sigma}], \mathcal{S}_{\hat{\Psi}}, Dom^*) \in r^{\hat{\Psi}}$, which is constructed according to construction rule (ii) for a dependency $\bar{\Gamma} \in P_i$ of a partition $P_i \in \mathcal{P}$ interfering with $\widehat{psec}_{prior}$. Because of Γ being in the partition P_j, which does *not* interfere with $\widehat{psec}_{prior}$, and because of $\bar{\Gamma}$ being in the partition P_i, which interferes with $\widehat{psec}_{prior}$, the partition P_i is *not* equal to the partition P_j. Hence, the non-implication $concl(\bar{\Gamma})[\bar{\sigma}] \not\models_{DB} prem(\Gamma)$ must hold under each constant substitution $\bar{\sigma}$ by construction of the partitioning $\mathcal{P}$. As $prem(\Gamma)$ is furthermore contained in the set $\mathcal{S}_{\hat{\Psi}}$ of all existentially quantified atoms *not* to be satisfied (if possible) by the ground operator, Lemma 5.4 guarantees that the tuple $grnd(concl(\bar{\Gamma})[\bar{\sigma}], \mathcal{S}_{\hat{\Psi}}, Dom^*)$ does *not* induce a DB-Interpretation satisfying $prem(\Gamma)$. As a consequence, this tuple $grnd(concl(\bar{\Gamma})[\bar{\sigma}], \mathcal{S}_{\hat{\Psi}}, Dom^*)$ does *not* induce a DB-Interpretation satisfying the stronger sentence $prem(\Gamma)[\sigma]$, either.

Moreover, the tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$ inducing a DB-Interpretation satisfying $prem(\Gamma)[\sigma]$ is *not* equal to a tuple $grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*) \in r^{\hat{\Psi}}$, which is constructed for a (possibly additional) potential secret Ψ of a cluster $C \in M_r^+$ according to construction rule (iii), either. If Ψ is a non-additional potential secret of $\widehat{psec}_{prior}$, the non-implication $\Psi \not\models_{DB} prem(\Gamma)$ is guaranteed as Γ is supposed to be in the partition P_j, which does *not* interfere with $\widehat{psec}_{prior}$. If Ψ is an additional potential secret, the validity of $\Psi \not\models_{DB} prem(\Gamma)$ is a direct consequence of the construction of admissible additional potential secrets ensuring that each dependency of *prior* does *not* interfere with an additional potential secret (cf. Definition 5.10). So, again considering that $prem(\Gamma)$ is contained in the set $\mathcal{S}_{\hat{\Psi}}$, Lemma 5.4 guarantees that the tuple $grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*)$ does *not* induce a DB-Interpretation satisfying $prem(\Gamma)$. As a consequence, this tuple $grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*)$ does *not* induce a DB-Interpretation satisfying the stronger sentence $prem(\Gamma)[\sigma]$, either.

Now, having excluded that the tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$ inducing a DB-Interpretation satisfying $prem(\Gamma)[\sigma]$ stems from rule (ii) or rule (iii) of the construction of the alternative instance $r^{\hat{\Psi}}$, the only possibility left is that $\boldsymbol{c}$ stems from construction rule (i). So, there is the ground atom $R(\boldsymbol{c}) \in weak(r, psec, prior)^+$ and hence the

original instance r also satisfies $prem(\Gamma)[\sigma]$, i.e., $\mathcal{I}_r \models_M prem(\Gamma)[\sigma]$, because of $\boldsymbol{c} \in r$. As $\mathcal{I}_r \models_M \Gamma$ is supposed to hold due to the precondition $\mathcal{I}_r \models_M prior$, there is also a tuple $\boldsymbol{d} \in r$ inducing a DB-Interpretation satisfying $concl(\Gamma)[\sigma]$.

Because of Γ being in the partition P_j *not* interfering with the policy $\widehat{psec}_{prior}$, there is *no* non-additional potential secret $\Psi \in \widehat{psec}_{prior}$, which shares a common constant unifier with $concl(\Gamma)$. Consequently, there is also *no* such non-additional potential secret Ψ in a cluster C of M_r^+ or in the subset $\widehat{psec}_{prior} \setminus \bigcup_{C \in M^+} C$ of those non-additional potential secrets *not* occurring in any cluster of M^+. Moreover, the construction of admissible additional potential secrets according to Definition 5.10 – ensuring that each dependency of *prior* does *not* interfere with an additional potential secret – guarantees that $concl(\Gamma)$ does *not* share a common constant unifier with any additional potential secret Ψ^A of a cluster C of M_r^+.

By further considering that $R(\boldsymbol{d})$ and $concl(\Gamma)$ share the common constant unifier $\boldsymbol{d}$ because of $R(\boldsymbol{d}) \models_{DB} concl(\Gamma)$, the non-implication $R(\boldsymbol{d}) \not\models_{DB} \Psi$ must hold for each (possibly additional) potential secret Ψ of a cluster $C \in M_r^+$ and for each (non-additional) potential secret Ψ of $\widehat{psec}_{prior} \setminus \bigcup_{C \in M^+} C$. Otherwise, $R(\boldsymbol{d})$ and Ψ and hence also Ψ and $concl(\Gamma)$ would share the common constant unifier $\boldsymbol{d}$, in contradiction with the insights gained above. According to Definition 5.11, this non-implication $R(\boldsymbol{d}) \not\models_{DB} \Psi$ for each Ψ of a cluster of M_r^+ and for each Ψ of $\widehat{psec}_{prior} \setminus \bigcup_{C \in M^+} C$ guarantees that the ground atom $R(\boldsymbol{d})$ is in $weak(r, psec, prior)^+$. Hence, the tuple $\boldsymbol{d}$ is in the alternative instance $r^{\hat{\Psi}}$ due to construction rule (i) and this instance $r^{\hat{\Psi}}$ accordingly satisfies $concl(\Gamma)[\sigma]$.

Subcase: Γ of the partition P_m, which interferes with $\widehat{psec}_{prior}$

To continue the proof of $\mathcal{I}_{r^{\hat{\Psi}}} \models_M prior$, suppose that m is defined and consider an arbitrary dependency $\Gamma \in prior$, which is in the partition $P_m \in \mathcal{P}$. This partition P_m interferes with the policy $\widehat{psec}_{prior}$, as the existence of a dependency $\bar{\Gamma} \in P_m$ with $concl(\bar{\Gamma})[\bar{\sigma}] \models_{DB} \hat{\Psi}$ is guaranteed under a constant substitution $\bar{\sigma}$ because of m being defined. Hence, Lemma 5.3 guarantees the existence of a non-additional potential secret $\Psi_{\text{prem}} \in \widehat{psec}_{prior}$ with $prem(\Gamma) \models_{DB} \Psi_{\text{prem}}$. For each tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$, which stems from a ground atom $R(\boldsymbol{c}) \in weak(r, psec, prior)^+$ according to construction rule (i), this ground atom $R(\boldsymbol{c})$ does *not* imply the potential secret Ψ_{prem} considered:

- if Ψ_{prem} is in a cluster $C \in M_r^+$ or in the set $\widehat{psec}_{prior} \setminus \bigcup_{C \in M^+} C$ of refused potential secrets, the non-implication $R(\boldsymbol{c}) \not\models_{DB} \Psi_{\text{prem}}$ is guaranteed by the construction of the positive knowledge $weak(r, psec, prior)^+$ of a weakened view according to Definition 5.11;

- if Ψ_{prem} is *neither* in a cluster of M_r^+ *nor* in the set $\widehat{psec}_{prior} \setminus \bigcup_{C \in M^+} C$ of refused potential secrets, then $\Psi_{\text{prem}} \notin (\widehat{psec}_{prior} \setminus \bigcup_{C \in M^+} C)$ guarantees that Ψ_{prem} is in a cluster $\bar{C} \in M^+$ and $\Psi_{\text{prem}} \notin \bigcup_{C \in M_r^+} C$ additionally guarantees $\mathcal{I}_r \not\models_M \bigvee_{\bar{\Psi} \in \bar{C}} \bar{\Psi}$ and hence also $\mathcal{I}_r \not\models_M \Psi_{\text{prem}}$. Consequently, the non-implication $R(\boldsymbol{c}) \not\models_{DB} \Psi_{\text{prem}}$ holds because of $\boldsymbol{c} \in r$ and $\mathcal{I}_r \not\models_M \Psi_{\text{prem}}$.

Hence, as each such tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$ stemming from construction rule (i) does *not* induce a DB-Interpretation satisfying the (weaker) potential secret Ψ_{prem}, it does *not* induce a DB-Interpretation satisfying the (stronger) sentence $prem(\Gamma)$, either, according to Lemma 3.1.

Next, consider an arbitrary tuple $grnd(concl(\bar{\Gamma})[\bar{\sigma}], \mathcal{S}_{\hat{\Psi}}, Dom^*) \in r^{\hat{\Psi}}$, which is constructed for a dependency $\bar{\Gamma} \in P_i$ of a partition $P_i \in \mathcal{P}$ with $i \neq m$ according to construction rule (ii). Because of $\Gamma \in P_m$, $\bar{\Gamma} \in P_i$ and $i \neq m$ and by construction of the partitioning $\mathcal{P}$, the non-implication $concl(\bar{\Gamma})[\bar{\sigma}] \not\models_{DB} prem(\Gamma)$ holds under the considered constant substitution $\bar{\sigma}$. As $prem(\Gamma)$ is moreover in the set $\mathcal{S}_{\hat{\Psi}}$ of all existentially quantified atoms *not* to be satisfied (if possible) by the ground operator, the tuple $grnd(concl(\bar{\Gamma})[\bar{\sigma}], \mathcal{S}_{\hat{\Psi}}, Dom^*)$ does *not* induce a DB-Interpretation satisfying $prem(\Gamma)$ according to Lemma 5.4.

Similarly, consider a tuple $grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*) \in r^{\hat{\Psi}}$, which is constructed for a (possibly additional) potential secret Ψ of a cluster $C \in M_r^+$ with $\Psi \not\models_{DB} prem(\Gamma)$ according to construction rule (iii). Due to $\Psi \not\models_{DB} prem(\Gamma)$ and by again considering that $prem(\Gamma)$ is in the set $\mathcal{S}_{\hat{\Psi}}$, the tuple $grnd(\Psi, \mathcal{S}_{\hat{\Psi}}, Dom^*)$ does *not* induce a DB-Interpretation satisfying $prem(\Gamma)$ according to Lemma 5.4.

Summing up, there is *no* tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$, which induces a DB-Interpretation satisfying the existentially quantified premise $prem(\Gamma)$ of a dependency $\Gamma \in P_m$. Accordingly, the constructed alternative instance $r^{\hat{\Psi}}$ satisfies each such dependency Γ of the considered partition P_m.

Subcase: Γ of a partition $P_i \in \mathcal{P}$ with $i \neq m$, which interferes with $\widehat{psec}_{prior}$

To finish the proof of $\mathcal{I}_{r^{\hat{\Psi}}} \models_M prior$, consider an arbitrary dependency $\Gamma \in prior$ and suppose that Γ is in a partition $P_i \in \mathcal{P}$ with $i \neq m$, which interferes with the extended and cleaned confidentiality policy $\widehat{psec}_{prior}$. If the alternative instance $r^{\hat{\Psi}}$ does *not* satisfy $prem(\Gamma)[\sigma]$ under *any* constant substitution σ, the validity of $\mathcal{I}_{r^{\hat{\Psi}}} \models_M \Gamma$ is a direct consequence. If the alternative instance $r^{\hat{\Psi}}$ instead satisfies $prem(\Gamma)[\sigma]$ under a constant substitution σ, i.e., $\mathcal{I}_{r^{\hat{\Psi}}} \models_M prem(\Gamma)[\sigma]$, there is a tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$, which induces a DB-Interpretation satisfying $prem(\Gamma)[\sigma]$.

Now, assume that $\boldsymbol{c}$ stems from a ground atom $R(\boldsymbol{c}) \in \mathit{weak}(r, \mathit{psec}, \mathit{prior})^+$, which is constructed according to rule (i) of the construction of $r^{\hat{\Psi}}$. As the partition P_i is supposed to interfere with $\widehat{\mathit{psec}}_{\mathit{prior}}$, Lemma 5.3 guarantees the existence of a potential secret $\Psi_{\text{prem}} \in \widehat{\mathit{psec}}_{\mathit{prior}}$ with $\mathit{prem}(\Gamma) \models_{DB} \Psi_{\text{prem}}$. Considering that $R(\boldsymbol{c}) \models_{DB} \mathit{prem}(\Gamma)[\sigma]$ is supposed to hold due to the assumption that $\boldsymbol{c}$ induces a DB-Interpretation satisfying $\mathit{prem}(\Gamma)[\sigma]$, the implication $R(\boldsymbol{c}) \models_{DB} \Psi_{\text{prem}}$ holds by transitivity because of

$$R(\boldsymbol{c}) \models_{DB} \mathit{prem}(\Gamma)[\sigma] \models_{DB} \mathit{prem}(\Gamma) \models_{DB} \Psi_{\text{prem}} \ .$$

But the potential secret Ψ_{prem} of the policy $\widehat{\mathit{psec}}_{\mathit{prior}}$, which is implied by $R(\boldsymbol{c})$, is either in the set $\widehat{\mathit{psec}}_{\mathit{prior}} \setminus \bigcup_{C \in M^+} C$ of refused potential secrets or in a cluster $C \in M_r^+$, whose corresponding disjunction $\bigvee_{\Psi \in C} \Psi$ is satisfied by the original instance r, as the assumption $R(\boldsymbol{c}) \in \mathit{weak}(r, \mathit{psec}, \mathit{prior})^+$ guarantees that $\boldsymbol{c}$ is in this instance r. Thus, the ground atom $R(\boldsymbol{c})$ is – in contradiction to the assumption – *not* qualified for being in the positive knowledge $\mathit{weak}(r, \mathit{psec}, \mathit{prior})^+$ of a weakened view according to Definition 5.11.

As the considered tuple $\boldsymbol{c} \in r^{\hat{\Psi}}$ – which is supposed to induce a DB-Interpretation satisfying $\mathit{prem}(\Gamma)[\sigma]$ – does *not* stem from construction rule (i), this tuple $\boldsymbol{c}$ is

- a tuple $\mathit{grnd}(\mathit{concl}(\bar{\Gamma})[\bar{\sigma}], \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*)$, which is constructed according to construction rule (ii) under a constant substitution $\bar{\sigma}$ substituting the universally quantified variables of the dependency $\bar{\Gamma}$ with constants stemming from
  - (possibly additional) potential secrets of clusters of M^+,
  - dependencies of the adversary's a priori knowledge *prior*, and
  - fresh constant symbols of the set Dom^*, or
- a tuple $\mathit{grnd}(\Psi, \mathcal{S}_{\hat{\Psi}}, \mathit{Dom}^*)$, which is constructed for a (possibly additional) potential secret Ψ of a cluster $C \in M_r^+$ according to construction rule (iii).

In both of these cases the tuple $\boldsymbol{c} = (c_1, \ldots, c_n)$ is constructed with the help of the (extended) ground operator and Lemma 5.4 hence guarantees that each constant symbol c_i of this tuple $\boldsymbol{c}$ stems from

- a (possibly additional) potential secret of a cluster of M^+,
- a dependency of the adversary's a priori knowledge *prior*, or
- the fresh constant symbols of the set Dom^*.

Reconsidering that the DB-Interpretation induced by the constant combination $\boldsymbol{c}$ is supposed to satisfy $prem(\Gamma)[\sigma]$, the constant substitution σ substitutes each variable of the existentially quantified premise $prem(\Gamma)$ with a constant symbol c_i of $\boldsymbol{c}$. As the dependency Γ is supposed to be in a partition $P_i \in \mathcal{P}$ with $i \neq m$, which interferes with the policy $\widehat{psec}_{prior}$, rule (ii) of the construction of the alternative instance $r^{\hat{\Psi}}$ guarantees that the tuple $grnd(concl(\Gamma)[\sigma], \mathcal{S}_{\hat{\Psi}}, Dom^*)$ is in $r^{\hat{\Psi}}$ – with σ being the above mentioned constant substitution induced by $\boldsymbol{c}$ – and thus the alternative instance $r^{\hat{\Psi}}$ satisfies $concl(\Gamma)[\sigma]$. This guarantees that $r^{\hat{\Psi}}$ also satisfies the dependency Γ, i.e., $\mathcal{I}_{r^{\hat{\Psi}}} \models_M \Gamma$. ♠

CHAPTER 6

Efficiency of the Weakening Algorithm

After the development of a generic algorithm computing weakened views on original database instances in order to provably enforce a considered confidentiality policy, an availability-maximizing instantiation of this approach has been proposed. This availability-maximizing instantiation has then even been extended to be also able to handle an adversary's a priori knowledge in the form of single premise tuple generating dependencies without losing its inference-proofness.

To demonstrate the practical efficiency these availability-maximizing weakening approaches can actually achieve, a prototype implementation has been developed and evaluated under different experiment setups. Each of these experiment setups systematically varies one of the parameters for an essentially random generation of input instances, which has a crucial impact on the runtime of the prototype implementation and possibly also on the generation of weakened views.

6.1 The Prototype Implementation

As known from the previous chapters, both the availability-maximizing weakening approach captured in Algorithm 4.1 as well as its extension handling a priori knowledge, which is formalized in Algorithm 5.1, still need to be instantiated with

a concrete implementation of a so-called notion of admissible indistinguishabilities to be fully specified. Thereby, such a notion of admissible indistinguishabilities essentially determines the structure disjunctions of weakened views should have. More precisely, such a concrete notion should induce algorithms determining the set of all admissible clusters over a given confidentiality policy (cf. Section 3.1.3) – which is essentially needed to construct the edges of an indistinguishability graph within the availability-maximizing approaches – as well as algorithms possibly (partly, within the extended approach) extending a maximum matching by pairing policy elements uncovered by this matching with admissible additional potential secrets (cf. Section 4.1.2 and Section 5.2.5).

Within the prototype implementation the notion of interchangeability, which is introduced in Section 4.2.1 and proved to be well-defined in case of the non-extended weakening approach according to Theorem 4.1, is employed as a concrete notion of admissible indistinguishabilities. Thereby, this proof that interchangeability is well-defined in case of the non-extended weakening approach also specifies an efficient and easy to implement algorithm for the construction of matching extensions, which comply with interchangeability. In case of the extended weakening approach a both sufficient and necessary condition for the constructibility of interchangeable and admissible additional potential secrets is provided by Theorem 5.1. The proof of this theorem is constructive in the sense that it sketches an efficient and easy to implement algorithm for the construction of interchangeable and admissible additional potential secrets whenever possible. Moreover, the declarative definition of interchangeability provided by Definition 4.5 naturally induces a straightforward algorithm to determine the set of all admissible clusters of size 2 over a given confidentiality policy in the spirit of a nested loop (self-)join algorithm known from relational databases [1, 75, 78].

During a preliminary experimental evaluation it turned out that for those (extended) confidentiality policies, which do *not* shrink much during the process of cleaning, the corresponding cleaned policies usually contain a quite large proportion of ground atoms. As a ground atom and a non-ground atom – i.e., an existentially quantified atom actually containing variables – can further *not* be interchangeable according to Definition 4.5, a cleaned confidentiality policy can first be partitioned into one subset containing all ground atoms of this policy and another subset containing all non-ground atoms of this policy. Subsequently, for both of these subsets all admissible clusters can be determined independently from the other subset without losing any admissible cluster to be constructed over the considered cleaned confidentiality policy. It hences makes sense to develop an improved algorithm for the construction of admissible clusters, which is specifically tailored to the characteristics of interchangeable ground atoms and which is therefore more efficient than the above mentioned naturally induced algorithm. Then,

this improved (more efficient) algorithm is used for the subset of all ground atoms of the cleaned policy and the naturally induced (less efficient, but applicable) algorithm is used for the remaining subset of all non-ground atoms.

This improved algorithm for the construction of all interchangeable pairs of ground atoms considers each possible single differing position $m \in \{1, \ldots, n\}$ of the ground atoms of arity n exactly once and for each $m \in \{1, \ldots, n\}$ the considered ground atoms are sorted lexicographically according to the order on their constant symbols, thereby *neglecting* the constant symbols at the m-th positions for this lexicographic order. Within such a (partially) ordered sequence the elements of each subset of pairwise interchangeable ground atoms – all only differing from each other in their m-th constant positions – follow one another consecutively. Thus, these subsets of pairwise interchangeable ground atoms are easy to identify within such a (partially) ordered sequence with a technique similar to the interleaved linear scans employed for the well-known sort-merge join algorithm [1, 75, 78].

In a worst-case scenario all ground atoms of an (extended and cleaned) confidentiality policy are pairwise interchangeable and hence form only one (sub-)set of pairwise interchangeable ground atoms. Then, for each pair of ground atoms an admissible cluster needs to be constructed and hence the runtime of this improved algorithm is – just as the runtime of the naturally induced algorithm – quadratic in the number of the considered ground atoms. But in typical scenarios the number of subsets of pairwise interchangeable ground atoms is significantly higher and hence only each pair of ground atoms, both of whose elements stem from the same of any of these (correspondingly smaller) subsets, needs to be considered. All other possible pairs of ground atoms, which do *not* form an admissible cluster, are instead *not* even considered by the improved algorithm and hence the runtime of this improved algorithm is usually significantly better than its quadratic worst-case complexity.

Moreover, a further optimization has been implemented for the cleaning of (extended) confidentiality policies consisting solely of ground atoms. While the general algorithm for cleaning a confidentiality policy proposed in Section 3.1.2 essentially aims at checking each pair of different potential secrets of such a policy for implication relationships, the cleaning of a policy solely consisting of ground atoms can be reduced to a removal of duplicate (i.e., semantically equivalent) policy elements, as there can *not* be any implication relationships within a set of pairwise (semantically) non-equivalent ground atoms (cf. Lemma 2.1). After sorting a considered set of ground atoms lexicographically, duplicates can be found easily as they follow one another consecutively within the ordered sequence.

Similarly, within Stage 2 of the weakening algorithm(s) the construction of the subset of those clusters, whose corresponding disjunctions are satisfied by a con-

sidered original database instance, also profits from an optimization for ground atoms: if a potential secret of a cluster is a ground atom, binary search can be employed within the sorted set of all ground atoms representing this database instance to decide on the satisfaction of this potential secret. After that, within the final construction of a weakened view, the search for those database tuples *not* implying the satisfaction of a weakening disjunction or a policy element to be refused – and hence occurring in the positive knowledge of this weakened view – is also optimized for ground atoms: if a database tuple (or, more precisely, its ground atom) can be found within the (sorted) list of those potential secrets it should *not* imply to occur in the positive knowledge, this tuple can immediately be excluded from the set of those database tuples qualified for being in the positive knowledge without employing an exhaustive search for implications.

To actually compute a (non-extended) clustering of policy elements on the basis of a maximum matching (cf. [50, 67, 77, 80]) determined on an indistinguishability graph, which represents a set of admissible clusters, the prototype benefits from an implementation of a maximum matching algorithm provided by the well-known "Boost"-library [41]. Although a maximum matching on a general (i.e., not necessarily bipartite) graph $G = (V, E)$ can be computed asymptotically best in $O(\sqrt{|V|} \cdot |E|)$ as known from [73, 89], common implementations of maximum matching algorithms as provided by the commercial "LEDA"-library [72] or the free "Boost"-library [41] prefer an algorithm performing in $O(|V| \cdot |E| \cdot \alpha(|E|, |V|))$ with $\alpha(|E|, |V|) \leq 4$ for any input of feasible size.

If an even faster matching computation is needed, the matching heuristic introduced in [71], which improves the seminal ideas proposed in [64] and performs in a time linear to the size of the graph, can be employed within the prototype. Although this matching heuristic guarantees the construction of a valid matching, it does *not* guarantee that this matching is actually maximum. Accordingly, the (extended) weakening algorithm might – in comparison with a maximum matching – need to construct more additional potential secrets to pair more policy elements uncovered by the resulting matching or might alternatively need to introduce more complete refusals to enforce these additionally uncovered policy elements. In both of these cases, this results in a loss of availability. But as evaluated in [71] (and confirmed by the experiments in Section 6.3), the employed matching heuristic usually loses only a negligible number of matching edges in relation to a maximum matching. Correspondingly, the usage of this heuristic within the prototype usually results only in a slight loss of availability.

To determine a partitioning $\mathcal{P}$ of an adversary's a priori knowledge *prior* within the extended weakening approach, the prototype uses an optimized variant of the algorithm proposed in Section 5.2.4. This (non-optimized) algorithm first

creates a partitioning graph $G = (V, E)$, whose edges essentially connect a pair of dependencies of *prior*, *if and only if* these dependencies need to be in the same partition. After that, each connected component of this partitioning graph G immediately induces a partition of $\mathcal{P}$.

Thereby, the construction of the edges of this partitioning graph is most costly in terms of runtime: to decide whether an edge between a pair of dependencies is to be constructed according to requirement (ii) of Definition 5.7, it is to be checked exhaustively whether both of these dependencies imply the same (arbitrary) potential secret of a considered extended and cleaned confidentiality policy under arbitrary constant substitutions. Although this check is, of course, only necessary for those pairs of dependencies, which are *not* already neighbored by an edge due to (the computationally less costly) requirement (iii) of Definition 5.7, it usually still needs to be performed for many possible pairs of dependencies.

However, the optimized variant of this partitioning algorithm – which is implemented within the prototype – tries to reduce the number of these costly checks considerably. Therefore, it takes into account that the results of potentially many of these costly checks might actually *not* affect the computed partitioning at all: if the partitioning graph already contains a path (of several edges) between a pair of dependencies, these dependencies are – due to the nature of connected components [65, 67] – guaranteed to be in the same connected component – and hence also in the same partition – anyway, independent of whether an edge directly connecting these dependencies is additionally added to the graph or *not*. Thereby, the problem of deciding on the existence of a path between a pair of vertices (dependencies) within a graph can be solved efficiently with the help of a so-called breadth-first search algorithm [65, 67] scanning all vertices of this graph, which can be reached from a specific vertex. In particular for input instances with large confidentiality policies and large sets of dependencies, which furthermore lead to partitioning graphs with large connected components, this optimization of the partitioning algorithm achieves considerable speedups.

All other basic subroutines of the prototype implementation are either explicitly specified in the course of the development of the weakening algorithm(s) or are naturally induced by the given (usually declarative) definitions of these subroutines. As all of the employed subroutines – including the above mentioned ones – obviously have a polynomial worst case complexity, both the non-extended as well as the extended availability-maximizing weakening algorithm also have a polynomial worst case complexity, provided that – just as interchangeability – the employed notion of admissible indistinguishabilities allows for the construction of efficient algorithms computing the set of all admissible clusters and efficient algorithms (partly) extending a computed maximum matching.

During a preliminary experimental evaluation of the prototype implementation the algorithms used for

- cleaning a confidentiality policy (*not* solely consisting of ground atoms),
- constructing a partitioning graph (to partition a priori knowledge),
- determining the subset of satisfied disjunction templates and
- creating the positive knowledge of a weakened view

turned out to be most costly in terms of runtime. As each of these algorithms – except for some optimizations – essentially relies on an exhaustive search for certain implication relationships, whose validities can be checked largely independent from each other, and as write accesses to common underlying data structures are typically rare, a natural approach to increase the performance of these algorithms with the help of modern hardware – usually offering multiple CPU cores – is to parallelize them. Although write accesses to common data structures are typically rare, care must nonetheless be taken to ensure that different threads running in parallel can write to these common data structures only in a synchronized way to avoid corruption of data and to moreover ensure that changes of data made by one thread are propagated to all other threads properly [63].

Although a parallelized implementation of a maximum matching algorithm would also be worthwhile for input instances leading to large indistinguishability graphs, such an implementation does *not* seem to be available for general (i.e., not necessarily bipartite) graphs. But for those input instances, for which the computation of a maximum matching is most costly in terms of runtime, one can still resort to the above mentioned matching heuristic to handle even large input instances.

6.2 The Experimental Setup

The prototype is implemented in Java 8, except for the above mentioned C++ implementation of the employed maximum matching algorithm provided by the "Boost"-library. All experiments were run under Ubuntu 14.04 on a machine with 2 CPU sockets, each of which is equipped with an "Intel Xeon E5-2690" CPU with 8 physical cores running at 2.9 GHz. Hence, a total number of 16 native CPU cores is available and – as each of these CPU cores can logically handle two threads simultaneously due to hyperthreading – the machine has a total number of 32 logical CPU cores. Although this machine is equipped with 64 GB of main memory, less than 10% of this available memory is actually needed to run the experiments sketched in the following.

For the experimental evaluation of the prototype implementation 5 different experiment setups are considered, each of which systematically varies the value of one of the parameters for an essentially random generation of input instances. To increase the significance of the experimental results, a total number of 100 input instances is generated for each considered instantiation of the generation parameters for these input instances. Correspondingly, each value of these experimental results is based on the average results of 100 repetitions of an experiment.

For each of the considered experiment setups database instances are generated over a database schema with 5 attributes, all ranging over a (fixed) domain with a total number of 20 constant symbols. Thereby, for each input instance instantiating the prototype implementation for a single run, an individual original database instance consisting of 1 000 000 (pairwise different) random database tuples is created over this database schema. But – despite this random creation of database instances – within those experiment setups also considering an adversary's a priori knowledge care is taken that each created database instance complies with each single premise tuple generating dependency of this a priori knowledge.

This compliance can be achieved by suitably adapting the well-known chase algorithm [1] in the sense that it is exhaustively checked which additional knowledge can be derived with the help of the given dependencies each time a new database tuple is added to a database instance under construction. For that purpose each dependency Γ of the given a priori knowledge, whose premise $prem(\Gamma)[\sigma]$ is satisfied by such a newly added database tuple under a constant substitution σ of the universally quantified variables of Γ, is considered. If the conclusion $concl(\Gamma)[\sigma]$ of this dependency is *not* satisfied by the database instance under the considered constant substitution σ of the universally quantified variables, a database tuple satisfying $concl(\Gamma)[\sigma]$ is constructed and additionally added to the database instance. Of course, such an additional tuple might then again be employed to derive further knowledge with the help of the considered dependencies. Hence, this chasing for additional database tuples by exhaustive forward chaining is repeated until the database instance under construction is consistent with all dependencies of the considered a priori knowledge. As a direct consequence, database instances constructed for experiment setups with a priori knowledge might have slightly more tuples than the target number of 1 000 000 database tuples.

Beside a database instance, each input instance instantiating the prototype implementation also contains a fully random confidentiality policy. Thereby, each experiment is evaluated under confidentiality policies of different sizes: the smallest of these policies consists of 10 000 (semantically) pairwise different potential secrets, after that the size of this policy is increased to 40 000 and then to 70 000

elements, and finally even a confidentiality policy with 100 000 pairwise different potential secrets is considered.

For the construction of these policy elements usually a subset of 12 constant symbols of the domain, over which the database instances are constructed, is available. This number of available constant symbols only deviates within Experiment 2, in which the number of available constant symbols is systematically varied from 10 to 22, thereby employing additional constant symbols *not* occurring in database tuples for those policies constructed over more than 20 constant symbols. The probability that a term of a potential secret is an existentially quantified variable (instead of a constant symbol) is usually 5% per term of a potential secret. Only within Experiment 1 this probability is systematically varied from 0% to 12%. Thereby, a probability of 0% leads to confidentiality policies consisting solely of ground atoms. During the construction of confidentiality policies *not* solely consisting of ground atoms, potential secrets *not* containing at least one constant symbol are discarded – to avoid the construction of weakest possible potential secrets, which are *not* feasible for the non-extended weakening approach (cf. Section 3.1.2) and usually undesirable (but generally feasible) for the extended weakening approach (cf. Section 5.3) – and a new random try for the construction of a valid potential secret is made.

At first glance, database tuples of arity 5, which are constructed over only 20 constant symbols, and corresponding confidentiality policies, whose active domain has a cardinality between 10 and 22, might look like "toy examples". But due to the nature of the employed interchangeability criterion, a suitably high number of admissible disjunction templates is only provided, if a considered confidentiality policy contains a lot of potential secrets structurally *not* differing much from each other. To achieve this under a fully random construction of potential secrets, both the arity of potential secrets (and hence also the arity of database tuples) as well as the number of available constant symbols is deliberately left low. To moreover guarantee that a suitably high number of database tuples induce DB-Interpretations satisfying a policy element, the number of available constant symbols for database tuples is correspondingly also left deliberately low.

Within each of the Experiments 3, 4 and 5 each input instance instantiating the extended weakening algorithm also contains an adversary's a priori knowledge, which usually consists of 1200 single premise tuple generating dependencies randomly constructed over the same domain of constant symbols as the database instances. Only in Experiment 3 this number of dependencies is systematically increased from 100 to 2500 in steps of 200 dependencies.

The probability that a term of a premise of a dependency is a universally quantified variable (instead of a constant symbol) is usually 15% per term of such a

Parameter	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
Database instance:					
number of tuples	1 000 000	1 000 000	$\geq$ 1 000 000	$\geq$ 1 000 000	$\geq$ 1 000 000
arity of tuples	5	5	5	5	5
number of constants	20	20	20	20	20
instance generation	fully random	fully random	random/chased	random/chased	random/chased
Confidentiality policy:					
number of potential secrets	$1, 4, 7, 10 \times 10^4$	$1, 4, 7, 10 \times 10^4$	$1, 4, 7, 10 \times 10^4$	$1, 4, 7, 10 \times 10^4$	$1, 4, 7, 10 \times 10^4$
number of constants	12	**from 10 to 22**	12	12	12
$\exists$ *quant. per term*	**from 0% to 12%**	5%	5%	5%	5%
policy generation	fully random	fully random	fully random	fully random	fully random
A priori knowledge:					
number of dependencies	–	–	**from 100 to 2500**	1200	1200
number of constants	–	–	as for instance	as for instance	as for instance
$\forall$ *quant. per term of premise*	–	–	15%	**from 5% to 29%**	15%
$\forall$ *quant. per term of conclusion*	–	–	10%	**5% less than above**	10%
$\exists$ *quant. per term of conclusion*	–	–	5%	5%	5%
dependency generation	–	–	random/corrected	random/corrected	random/corrected
Parallelization:					
number of threads	64	64	64	64	**from 1 to 25**

Figure 6.1: Parameters of experiments (varying parameter values in boldface)

premise, as tuple generating dependencies often have quite a high proportion of variables. Similarly, the probability that a term of a conclusion of a dependency is a (randomly chosen) universally quantified variable of the premise of this dependency is usually 10% per term of such a conclusion. Thereby, care is taken that the conclusion of such a dependency can *not* contain a higher number of universally quantified variables than the premise of this dependency, as each of these variables can occur only once in the premise and in the conclusion of a single premise tuple generating dependency (cf. Definition 5.1). Similarly to the construction of potential secrets, both the premise and the conclusion of a dependency must moreover contain at least one constant symbol to avoid the construction of extended confidentiality policies (cf. Definition 5.5) containing a weakest possible potential secret. In case that a randomly constructed dependency violates one of these requirements, this dependency is discarded and a new random try for the construction of a valid dependency is made.

Deviating from the above mentioned probabilities, within Experiment 4 the probability that a term of a premise of a dependency is a universally quantified variable ranges from 5% to 29% in steps of 2 percentage points and the probability that a term of a conclusion of a dependency is a (randomly chosen) universally quantified variable of the premise of this dependency is 5 percentage points less than the probability that a term of the premise of this dependency is a universally quantified variable. Without any exceptions, the probability that a term of a conclusion of a dependency is an existentially quantified variable – if it is *not* already chosen to be a universally quantified variable – is always 5% per term.

As some subroutines of the weakening algorithm(s) are parallelized as documented in Section 6.1, the prototype should also be instantiated with the number of threads to run in parallel. If this number is *not* explicitly set, the prototype follows the widely used rule of thumb that the number of threads should be twice the number of available CPU cores – in the hope that each CPU core can still be fully utilized, even if some threads need to wait until others release locks on some resources (cf. [63, 87]), without increasing the costs arising due to CPU context switches (cf. [87]) too much. On the above described machine with a total number of 32 logical CPU cores this rule of thumb leads to a default value of 64 threads to run in parallel. Unless otherwise stated, this default value is used for each experiment setup except for Experiment 5: this last experiment aims at evaluating the effectiveness of parallelization and therefore systematically varies the number of threads to run in parallel from 1 to 25 in steps of 2 threads.

For convenience, the parameters used to generate the input instances for the different experiment setups are summarized in the table given in Figure 6.1. Thereby, for each experiment its systematically varied parameter is printed in boldface.

6.3 Experimental Evaluation

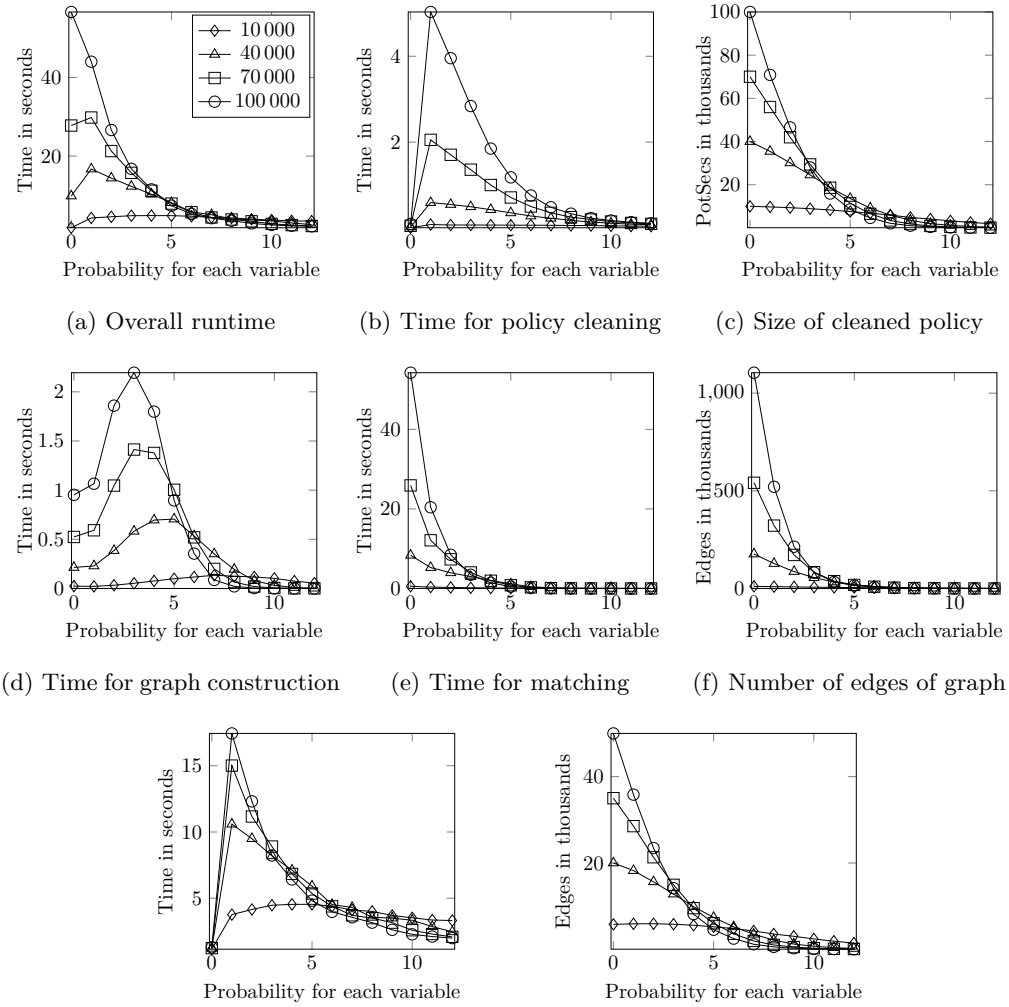
In the following, each of the above mentioned experiment setups is evaluated in detail. For that purpose, selected curves are presented for each experiment setup and then interpreted against the background of the prototype implementation.

Experiment 1. Within Experiment 1 the probability that a term of a potential secret of a confidentiality policy is an existentially quantified variable is systematically varied from 0% to 12% per term to measure the impact of existential quantification of policy elements on the runtime of the prototype implementation. For confidentiality policies consisting solely of ground atoms – i.e., the probability for existential quantification is 0% – the results depicted in Figure 6.2 indicate that the computation of maximum matchings, whose runtime is given in Figure 6.2(e), clearly dominates the overall runtime given in Figure 6.2(a).

But when considering confidentiality policies *not* solely consisting of ground atoms, the runtime for the computation of maximum matchings on indistinguishability graphs strongly declines with an increasing probability for existential quantification within the elements of a confidentiality policy. The reason for this is documented in Figure 6.2(c) and in Figure 6.2(f): an increasing number of existentially quantified variables occurring in policy elements results in a shrinking of cleaned confidentiality policies, over which indistinguishability graphs are constructed. This in turn results in considerably smaller – and thus much faster to solve – input instances for the maximum matching algorithm.

In contrast to the computation of maximum matchings, the runtime for the cleaning of confidentiality policies given in Figure 6.2(b) and the runtime for Stage 2 of the weakening algorithm – i.e., determining those clusters, whose corresponding disjunctions are satisfied by a considered database instance, and the construction of a weakened view – given in Figure 6.2(g) both rise sharply with the introduction of existential quantification within confidentiality policies. While the runtimes of these subroutines strongly benefit from optimizations implemented for the special case of ground atoms (cf. Section 6.1), the benefit of these optimizations decreases considerably in the case of Stage 2 of the weakening algorithm and vanishes completely in the case of cleaning a confidentiality policy with the introduction of existential quantification. Similar, but less severe effects related to optimizations for ground atoms can be observed for the construction of indistinguishability graphs, whose runtime is depicted in Figure 6.2(d).

But in general the runtimes of all of these subroutines nonetheless tend to decrease with an increasing probability for existential quantification, which obviously re-



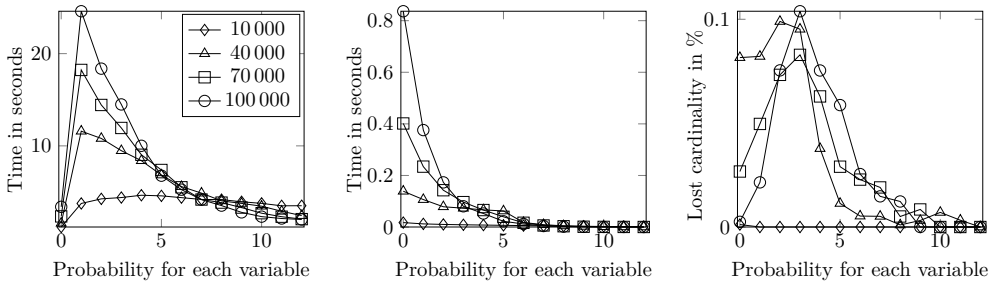
(a) Overall runtime (b) Time for policy cleaning (c) Size of cleaned policy

(d) Time for graph construction (e) Time for matching (f) Number of edges of graph

(g) Runtime of Stage 2 (h) Size of matching extension

Figure 6.2: Experiment 1: Varying number of variables in policies

sults in an increasing number of implication relationships between pairs of policy elements. In case of the algorithm cleaning a confidentiality policy such a higher number of implication relationships leads to more policy elements to be removed in early iterations of this cleaning algorithm and each of these removed policy elements does *not* need to be considered in subsequent iterations of this algorithm any more. The resulting decrease of the cardinalities of cleaned confidentiality policies (cf. Figure 6.2(c)) then naturally leads to faster constructions of indis-



(a) Overall runtime (b) Time for matching heuristic (c) Quality of heuristic

Figure 6.3: Experiment 1 with matching heuristic

tinguishability graphs (cf. Figure 6.2(d)) and to smaller matching extensions (cf. Figure 6.2(h)) – whose construction time is negligible and hence *not* further considered in Figure 6.2. Due to these smaller matching extensions, within Stage 2 of the weakening algorithm both the construction of the subset of those clusters of matching extensions corresponding to satisfied disjunctions as well as the search for those database tuples *not* implying the satisfaction of a weakening disjunction – and hence occurring in the positive knowledge of a weakened view – then become less complex and hence faster (cf. Figure 6.2(g)).

As already proposed in Section 6.1, a matching heuristic can be employed instead of an exact maximum matching solver to improve the runtime for matching computations. Thus, an adaption of the prototype implementation, which relies on this proposed matching heuristic, has also been evaluated on the basis of the input instances generated for Experiment 1. As can be seen in Figure 6.3(b), this matching heuristic improves the runtime for matching computations significantly in comparison to the exact maximum matching solver provided by the "Boost"-library, whose performance is shown in Figure 6.2(e).

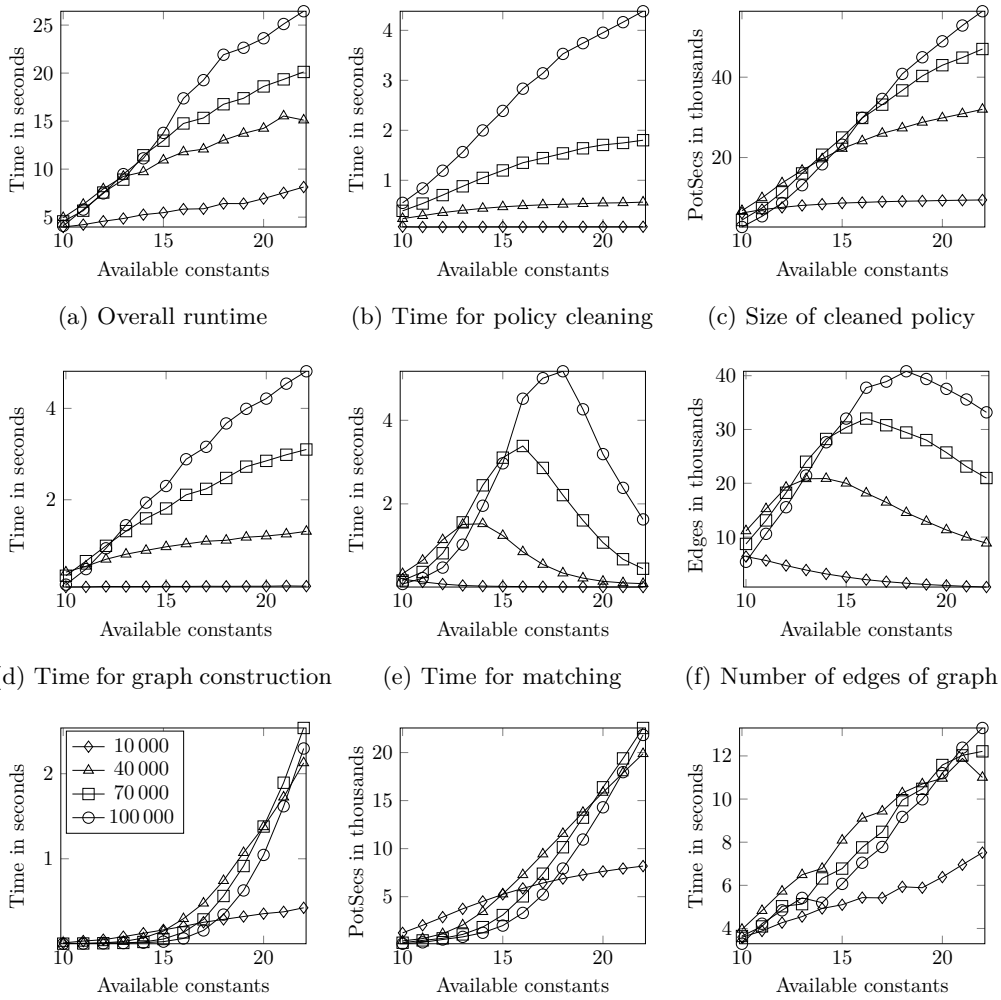
This is in particular true for low probabilities for existential quantification within the elements of a confidentiality policy, which lead to large indistinguishability graphs (cf. Figure 6.2(f)), on which exact computations of maximum matchings are most costly in terms of runtime. As evaluated above, the time needed for matching computations dominates the overall runtime of the prototype implementation in these scenarios and correspondingly the usage of the matching heuristic improves this overall runtime of the prototype implementation significantly as shown in Figure 6.3(a). Especially for confidentiality policies consisting solely of ground atoms, for which optimizations are implemented for several subroutines of the prototype implementation as outlined above, the usage of the matching heuris-

tic leads to overall runtimes of less than 4 seconds – even for those input instances with confidentiality policies of 100 000 potential secrets to be clustered.

As clarified in Section 6.1, the employed matching heuristic constructs only valid matchings, but does *not* guarantee that these matchings are actually maximum. Hence, the quality of matchings determined by this heuristic can be measured by observing how much smaller the cardinalities of these heuristically determined matchings are in comparison with the cardinalities of those matchings computed by an exact matching solver on the same indistinguishability graphs. As documented in Figure 6.3(c), this loss of cardinality does *not* significantly exceed 0.1% in average. Accordingly, the number of additional potential secrets, which the weakening algorithm needs to construct to pair each policy element uncovered by a matching, does usually *not* notably increase when using the matching heuristic. Thus, the usage of this heuristic results only in a negligible loss of availability and might hence be a great compromise for scenarios leading to large indistinguishability graphs and requiring a maximum of efficiency.

Experiment 2. To measure the impact of an increasing number of constant symbols occurring in randomly constructed confidentiality policies, Experiment 2 is set up and varies the number of available constant symbols systematically from 10 to 22. Thereby, a first look at Figure 6.4(a) seems to suggest that a higher number of constant symbols immediately increases the overall runtime of the prototype implementation. But a closer look at the reasons for this increase of runtime reveals that this increase is also closely related to the random construction of confidentiality policies.

As a first step to attain this insight, consider that in case of the larger confidentiality policies with 40 000, 70 000 and 100 000 elements the sizes of the corresponding cleaned confidentiality policies increase almost proportionally with the number of available constant symbols as shown in Figure 6.4(c). With higher numbers of available constants the probability that randomly chosen constants occurring at certain parameter positions of a pair of potential secrets are equal becomes smaller and, correspondingly, the probability that there is an implication relationship between these potential secrets, which leads to a removal of one of these policy elements during the cleaning of the policy, also decreases (cf. Lemma 2.1). As an immediate consequence, the time needed for cleaning these confidentiality policies increases with the number of available constant symbols as documented in Figure 6.4(b), as each potential secret of a confidentiality policy, which remains in this policy after one check for an implication relationship, also needs to be considered in subsequent iterations of the cleaning algorithm checking for the validity of other implication relationships.



(a) Overall runtime (b) Time for policy cleaning (c) Size of cleaned policy

(d) Time for graph construction (e) Time for matching (f) Number of edges of graph

(g) Time matching extension (h) Number additional PotSecs (i) Runtime of Stage 2

Figure 6.4: Experiment 2: Varying number of constants in policy

In case of the small confidentiality policies with only 10 000 elements the increase of the size of cleaned confidentiality policies and the corresponding increase of the runtime for the process of cleaning are less severe. The probability for overlapping constant symbols is – in comparison with the above considered large confidentiality policies – already low for small domains of available constant symbols due to choosing a comparatively small random subset of policy elements from the set of all constructible potential secrets. Accordingly, even for these small domains of constant symbols the number of policy elements, which are removed during the

process of cleaning, is already comparatively low and the size of a confidentiality policy does hence *not* shrink that much during this process of cleaning.

With increasing cleaned confidentiality policies the time needed for the construction of indistinguishability graphs also increases as shown in Figure 6.4(d): larger cleaned policies lead to indistinguishability graphs with more vertices and for each pair of these vertices the admissibility of an edge is to be examined. The number of these admissible edges, which is captured in Figure 6.4(f), first also increases with the number of vertices. But then, after some turning point, it steadily decreases, as the number of potential secrets differing from another potential secret in only one single constant symbol – thereby inducing edges, which are admissible according to interchangeability – decreases with an increasing number of available constant symbols, over which potential secrets are constructed randomly. Correspondingly, the runtime of the maximum matching algorithm first increases and then decreases as documented in Figure 6.4(e), as the number of possible ways to actually construct maximum matchings is more and more reduced due to an increasing number of vertices in relation to a decreasing number of edges.

As a further consequence of these reduced possibilities to construct maximum matchings, the number of original policy elements uncovered by maximum matchings increases with the number of available constant symbols and hence the number of additional potential secrets needed to pair uncovered original policy elements rises as shown in Figure 6.4(h). This additional expense for the construction of matching extensions is also reflected in the time needed for the construction of these matching extensions as depicted in Figure 6.4(g). As each additional potential secret requires additional knowledge to be distorted, extended matchings with a large proportion of additional potential secrets also reduce the achieved level of availability. Hence, the notion of interchangeability, which is responsible for this high number of additional potential secrets under random confidentiality policies and increasing domains of constant symbols as outlined above, does *not* seem to be an appropriate notion of admissible indistinguishabilities – prudently balancing confidentiality and availability – for scenarios considering (virtually) random confidentiality policies over larger domains.

Reconsidering the increasing cardinalities of extended matchings – due to both higher cardinalities of cleaned confidentiality policies and higher numbers of unmatched elements of these policies – Stage 2 of the weakening algorithm also becomes more complex with an increasing number of available constant symbols as shown in Figure 6.4(i): both the construction of the subset of those edges of a matching extension corresponding to satisfied disjunctions as well as the search for those database tuples *not* implying the satisfaction of a weakening disjunction corresponding to an edge of a matching extension – and hence occurring in the

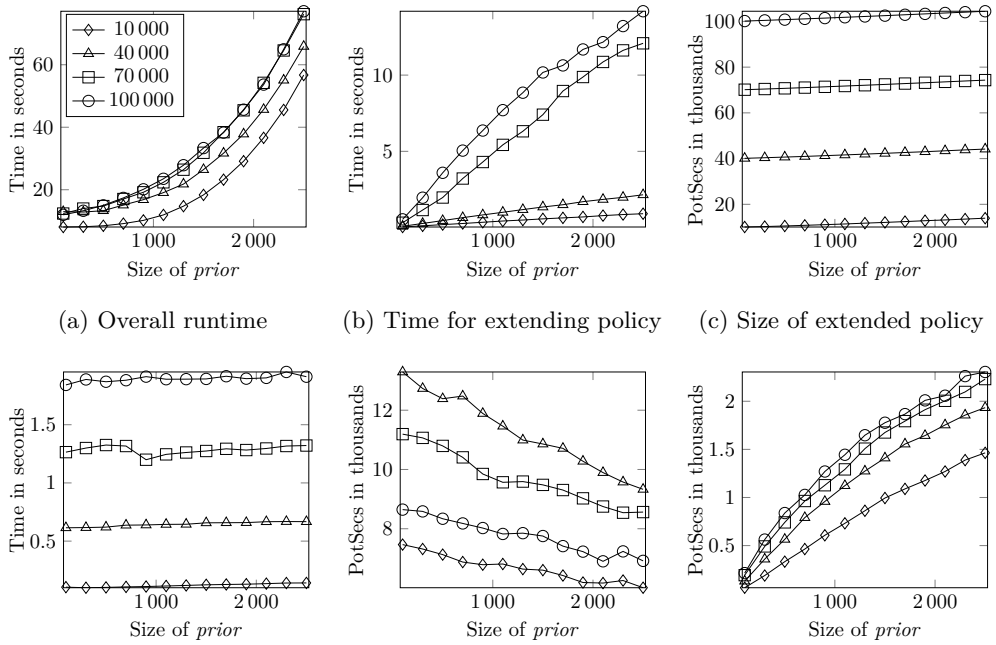
positive knowledge of a weakened view – become more time consuming with larger extended matchings.

Experiment 3. To also evaluate the extended weakening algorithm, an adversary is from now on supposed to have some a priori knowledge in the form of single premise tuple generating dependencies. In order to measure the impact of the size of an adversary's a priori knowledge, Experiment 3 systematically varies the number of considered dependencies from 100 to 2500. Thereby, the graphs given in Figure 6.5(a) immediately reveal that the overall runtime of the prototype implementation considerably increases with the number of dependencies, but still remains practically feasible.

The first step of the extended weakening algorithm is to extend a given confidentiality policy by the existentially quantified premise and the existentially quantified conclusion of each dependency interfering with the current (possibly already partly extended) policy until a fixpoint is reached in the sense that *no* further interferences can be found. According to the results presented in Figure 6.5(c), this extension leads to a linear increase of (extended) confidentiality policies in approximately the size of the considered a priori knowledge. Similarly, the time needed to extend a confidentiality policy also increases almost linearly with the number of considered dependencies, but the gradient of this increase also depends on the size of the considered confidentiality policy, as in a worst-case scenario each possible combination of a policy element and a dependency needs to be considered.

With an increasing size of extended confidentiality policies, the time needed to clean these policies also increases slightly, but remains more or less constant as shown in Figure 6.5(d). In contrast, the sizes of cleaned confidentiality policies sketched in Figure 6.5(e) shrink considerably: due to the quite high proportion of universally quantified variables within dependencies, existentially quantified premises and conclusions added to extended policies often contain a high proportion of existentially quantified variables in comparison with other policy elements and are hence often implied by quite a high number of policy elements, which are then to be removed during the process of cleaning. This is in particular true for confidentiality policies of large size: according to Figure 6.5(e), cleaned policies resulting from these large (uncleaned) policies are often even smaller than cleaned policies resulting from smaller (uncleaned) policies.

Although the time needed to determine all policy elements to be refused – because of being implied by the conclusion of a dependency under an arbitrary constant substitution – is negligible and hence *not* further considered, the number of these



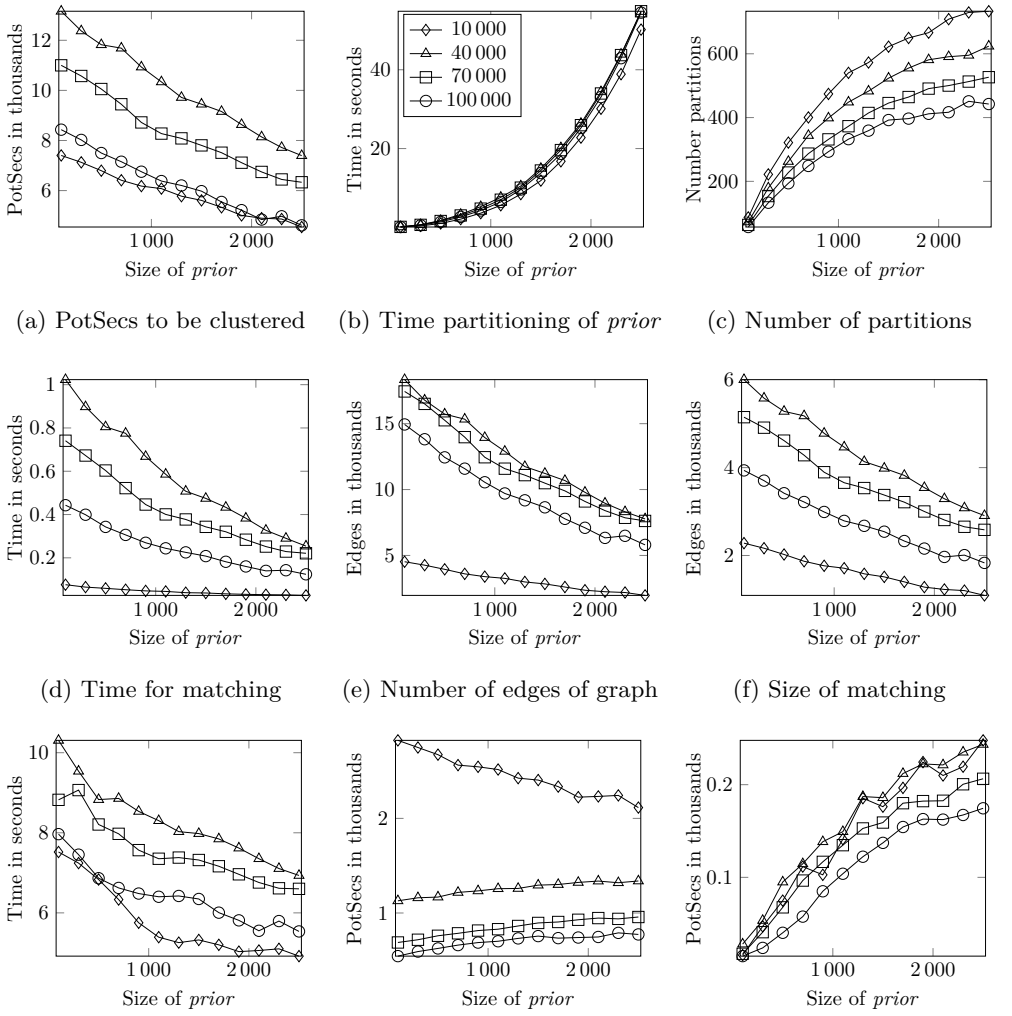
(a) Overall runtime (b) Time for extending policy (c) Size of extended policy

(d) Time for cleaning the policy (e) Size of cleaned policy (f) PotSecs to be refused

Figure 6.5: Experiment 3: Varying number of dependencies (1)

refused policy elements is of interest, as the primary goal of the developed weakening algorithm is to enforce confidentiality requirements with the help of weakening disjunctions. As shown in Figure 6.5(f), the number of refused policy elements grows almost proportionally with the number of considered dependencies. This is a promising result, as for each dependency Γ interfering with a considered extended and cleaned confidentiality policy there is a potential secret in this policy, which is implied by the (existentially quantified) conclusion of Γ – and is thus to be refused anyway – and furthermore a substantial share of the considered dependencies interferes with the considered policies (cf. Figure 6.5(c)). In comparison with Figure 6.6(a) it becomes clear that the majority of the elements of an extended and cleaned policy (cf. Figure 6.5(e)) is still protected with the help of weakening disjunctions under the considered experiment setup.

According to Figure 6.6(b) the time needed to determine a partitioning of an adversary's a priori knowledge clearly increases with the number of dependencies contained in such an a priori knowledge. Thereby, the size of a considered (cleaned) confidentiality policy does *not* influence this runtime much. Additionally considering the overall runtime of the prototype implementation given in Figure 6.5(a),



(a) PotSecs to be clustered (b) Time partitioning of *prior* (c) Number of partitions

(d) Time for matching (e) Number of edges of graph (f) Size of matching

(g) Runtime of Stage 2 (h) Additional PotSecs (i) Refused unmatched PotSecs

Figure 6.6: Experiment 3: Varying number of dependencies (2)

it becomes clear that – in particular for a priori knowledges of large size – the time needed to partition an adversary's a priori knowledge clearly dominates the overall runtime of the prototype implementation. The number of partitions also grows with an increasing number of dependencies as shown in Figure 6.6(c). But this growth is clearly below a linear increase, as a higher number of dependencies increases the probability of implication relationships – between conclusions and premises of dependencies as well as between conclusions of dependencies and

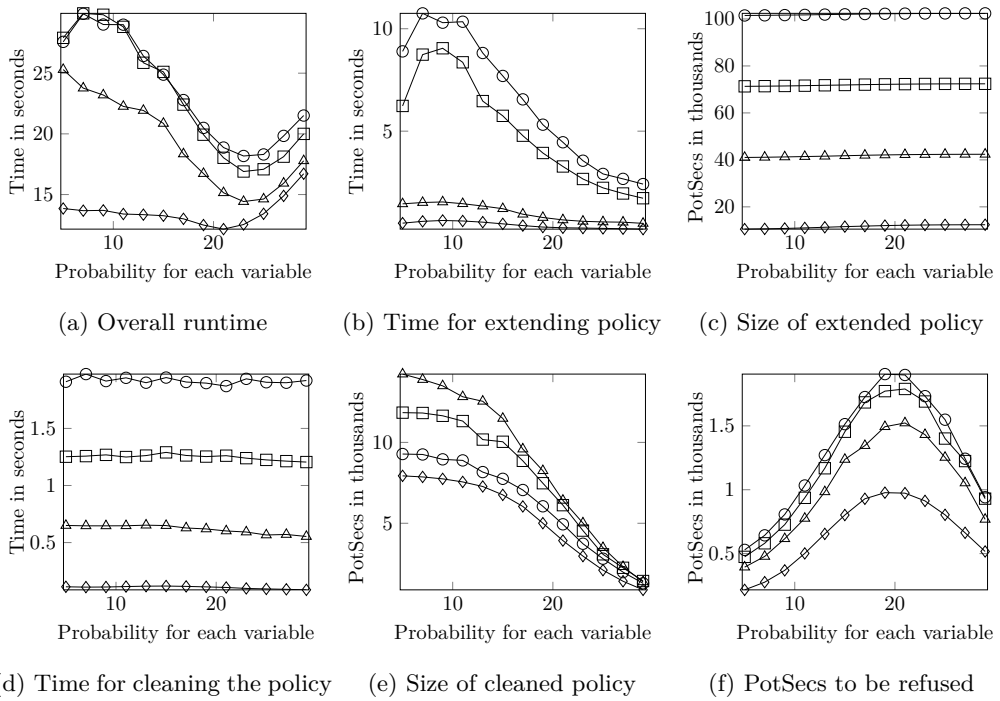
policy elements – and hence leads to partitions of larger cardinality.

Similar to the previous experiment setups, the time needed to compute maximum matchings (cf. Figure 6.6(d)) depends on the size of the indistinguishability graphs (cf. Figure 6.6(e)). As the graphs resulting from this experiment setup are comparatively small due to the small cardinalities of cleaned confidentiality policies (cf. Figure 6.5(e)), the impact of maximum matching computations on the overall runtime of the prototype implementation is nearly negligible. In correspondence with the decreasing number of edges of indistinguishability graphs (cf. Figure 6.6(e)), the size of maximum matchings also decreases with an increasing number of dependencies according to Figure 6.6(f). This, in turn, results in a runtime of Stage 2 of the weakening algorithm which decreases with the size of maximum matchings as well as depicted in Figure 6.6(g).

As known from Section 5.4, the interchangeability criterion might *not* allow for the construction of an admissible additional potential secret for each policy element uncovered by a maximum matching, as each constructible additional potential secret might interfere with a dependency of a considered a priori knowledge due to a common constant unifier. In this case, it is necessary to refuse such an uncovered policy element completely. But reconsidering the construction goals of the weakening approach postulated in Section 1.3.1, confidentiality requirements should usually be enforced with the help of weakening disjunctions – instead of refusals – whenever possible. A comparison of Figure 6.6(h) and Figure 6.6(i) reveals that most of the potential secrets uncovered by a maximum matching can indeed be enforced with the help of weakening disjunctions, as the number of uncovered policy elements, which need to be refused, is considerably smaller than the number of constructed additional potential secrets.

Experiment 4. To next evaluate the impact of universal quantification within dependencies of an adversary's a priori knowledge, Experiment 4 systematically varies the probability that a term of a premise of a dependency is a universally quantified variable from 5% to 29%. Moreover, the probability that a term of a conclusion of a dependency is a universally quantified variable (also occurring in the premise) is always 5 percentage points less than the probability that a term of a premise is a universally quantified variable.

According to Figure 6.7(b) the time needed to determine an extended confidentiality policy essentially decreases with a higher probability for universal quantification: a higher number of universally quantified variables within a dependency increases the probability that this dependency interferes with a potential secret of a considered confidentiality policy and for each dependency *no* further checks



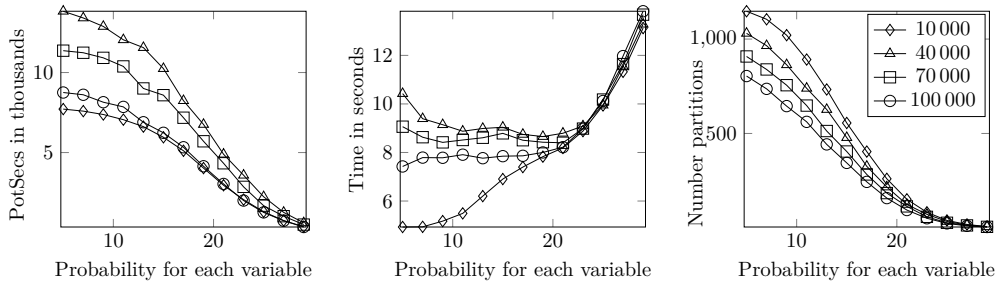
(a) Overall runtime (b) Time for extending policy (c) Size of extended policy

(d) Time for cleaning the policy (e) Size of cleaned policy (f) PotSecs to be refused

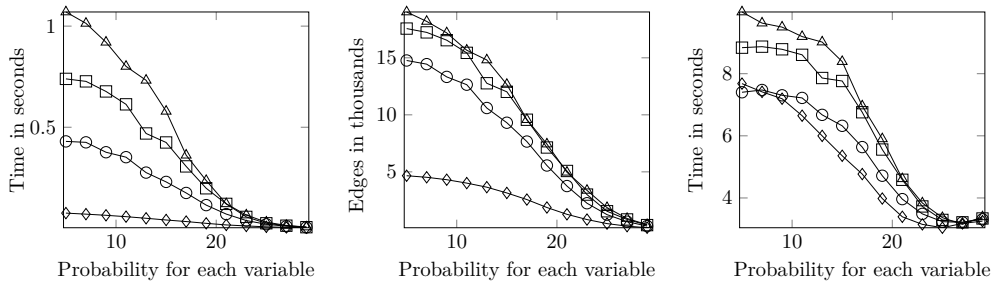
Figure 6.7: Experiment 4: Varying number of "∀-variables" in *prior* (1)

for interference with a confidentiality policy need to be performed as soon as an interference relationship with a single element of this confidentiality policy is successfully found. Correspondingly, the size of extended confidentiality policies slightly increases with a higher probability for universal quantification as documented in Figure 6.7(c), as higher probabilities for interferences lead to more dependencies interfering with a considered confidentiality policy.

Due to this increasing number of existentially quantified premises and conclusions of dependencies within extended confidentiality policies and due to an increasing proportion of (existentially quantified) variables within these premises and conclusions, the number of (other) policy elements implying such an existentially quantified premise or conclusion also increases with the probability for universal quantification. As a direct consequence, the size of cleaned and extended confidentiality policies shrinks with a higher probability for universal quantification as shown in Figure 6.7(e). Similar to Experiment 3, this is again in particular true for confidentiality policies of large size and cleaned policies resulting from larger (uncleaned) policies are hence often even smaller than cleaned policies resulting from smaller (uncleaned) policies.



(a) PotSecs to be clustered (b) Time partitioning of *prior* (c) Number of partitions

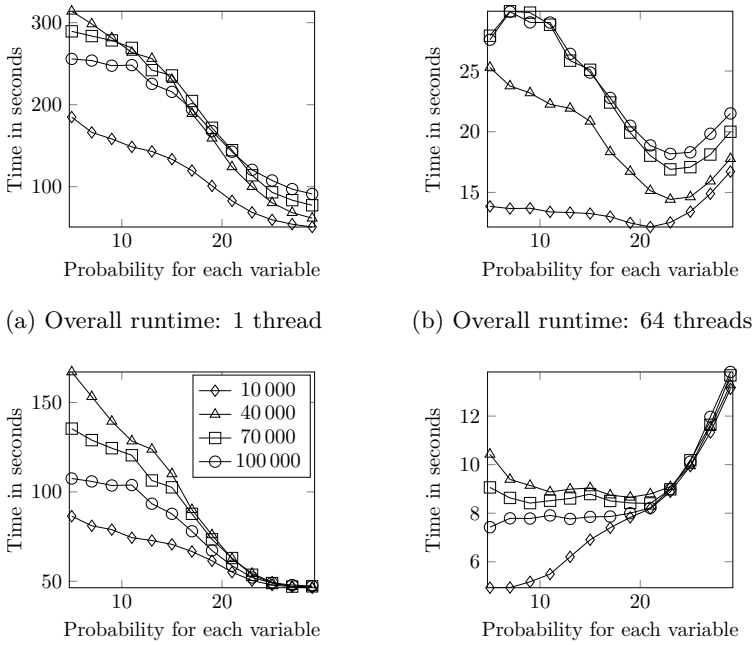


(d) Time for matching (e) Number of edges of graph (f) Runtime of Stage 2

Figure 6.8: Experiment 4: Varying number of "∀-variables" in *prior* (2)

As a further consequence of an increasing number of universally quantified variables within conclusions of dependencies, the probability that such a conclusion implies a policy element under an arbitrary constant substitution of its universally quantified variables also grows. As shown in Figure 6.7(f), this first leads to an increasing number of policy elements to be refused in spite of shrinking sizes of cleaned confidentiality policies (cf. Figure 6.7(e)), but then, after some turning point, this growth of the number of policy elements to be refused stops and the number of refused policy elements then decreases with the size of cleaned confidentiality policies. But overall, this number of refused policy elements does again *not* seem to influence the number of policy elements to be protected by weakening disjunctions much: a comparison between Figure 6.7(e) and Figure 6.8(a) immediately indicates that the number of (cleaned) policy elements to be clustered essentially depends on the size of cleaned confidentiality policies.

Now considering that a higher number of universally quantified variables within conclusions of dependencies also increases the probability that such a conclusion implies – under a certain constant substitution – the premise of another dependency or a (cleaned) policy element, it seems quite natural that the number of



(a) Overall runtime: 1 thread

(b) Overall runtime: 64 threads

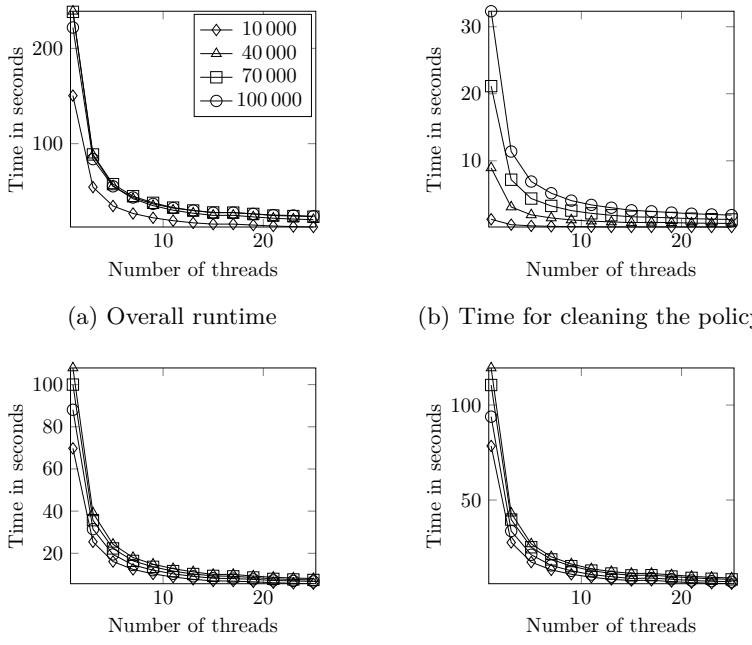
(c) Partitioning: 1 thread

(d) Partitioning: 64 threads

Figure 6.9: Experiment 4: Single-threaded vs. multi-threaded

partitions, which result from the partitioning of an adversary's a priori knowledge according to Definition 5.7, decreases with an increasing probability for universal quantification as shown in Figure 6.8(c). In contrast, it does *not* seem to be reasonable that the runtime for the partitioning of an adversary's a priori knowledge increases with the probability for universal quantification as depicted in Figure 6.8(b), as easier (and hence usually faster) to find implication relationships of the above mentioned kind should result in a decrease of runtime.

Considering Figure 6.9(c), such a decrease of runtime for the partitioning of an adversary's a priori knowledge can indeed be observed, if the prototype implementation is run single-threaded. To confirm that the above mentioned increase of runtime is caused by a growing overhead due to synchronization effects between different threads running in parallel, reconsider that for each found implication relationship, which actually leads to a new edge within the partitioning graph, the thread adding this new edge to the partitioning graph must have exclusive (write) access to the data structure managing the graph – and all other threads wishing to access this data structure are correspondingly blocked. Hence, an increasing number of write accesses to the partitioning graph, which immediately results



(a) Overall runtime

(b) Time for cleaning the policy

(c) Time partitioning of *prior*

(d) Runtime of Stage 2

Figure 6.10: Experiment 5: Varying number of threads

from the construction of larger partitions of dependencies, leads to more blocking of threads and hence results in a runtime increasing with the probability for universal quantification within dependencies. But nonetheless, both the performance of the partitioning of an adversary's a priori knowledge as well as the performance of the overall algorithm still profit greatly from the merits of parallelization as demonstrated impressively in Figure 6.9.

Similarly to the previous experiment setups, the size of cleaned confidentiality policies (cf. Figure 6.7(e)) influences the size of indistinguishability graphs (cf. Figure 6.8(e)), on which then the runtime for the computation of maximum matchings on these graphs depends. As also known from the previous experiment setups, smaller indistinguishability graphs usually lead to smaller (partly extended) maximum matchings and hence the runtime of Stage 2 of the weakening algorithm also decreases with the probability for universal quantification within dependencies according to Figure 6.8(f).

Experiment 5. To finally evaluate the effectiveness of parallelization, Experiment 5 systematically varies the number of threads running in parallel from 1 to

25. According to the Figures 6.10(b), 6.10(c) and 6.10(d), all parallelized subroutines profit nearly optimally from an increasing number of threads, as a doubling of the number of threads running in parallel nearly halves the runtimes of these subroutines. Moreover, the overall runtime of the prototype implementation given in Figure 6.10(a) behaves similarly well under an increasing number of threads, although this overall runtime also includes the runtimes of all non-parallelized subroutines. This insight confirms that actually those subroutines of the weakening algorithm(s), which have the most crucial impact on the overall runtime of the prototype implementation, have been chosen to be parallelized.

CHAPTER 7

Conclusion & Future Work

Now that the weakening approach has been developed successfully and the practical efficiency of its availability-maximizing instantiation has experimentally been confirmed, it is finally time to conclude this thesis. To this end, the main novel contributions of this thesis are now summarized, followed by a discussion how future work might further extend these contributions.

7.1 Contributions of this Thesis

Motivated by the emerging challenge of confidentiality preserving data publishing, which can *not* be implemented satisfactorily on the basis of those approaches traditionally used to enforce confidentiality requirements, Chapter 1 explores how confidentiality preserving data publishing can be realized within the framework of Controlled Interaction Execution. In contrast to traditional approaches in the spirit of access control, which operate on the level of raw data, this framework of Controlled Interaction Execution aims at suitably confining an adversary's possible gain of information such that this adversary is provably *not* able to infer a piece of knowledge to be kept confidential by employing his reasoning capabilities.

As the exploration in Chapter 1 comes to the conclusion to develop a novel approach replacing confidentiality compromising knowledge of a considered database instance by weaker disjunctive knowledge, Chapter 2 introduces a logic-based framework to provide a basis for the development of this novel approach. Thereby, this framework relies on a restricted but expressive subclass of first-order logic, which meets the requirement that the validity of implication relationships can be decided efficiently without costly general theorem proving. These restrictions thus allow for the construction of a computationally efficient approach, which is able to handle even large input instances resulting from steadily growing collections of data. Subsequently, a first basic weakening approach is presented under the supposition that only a simplified kind of input instances is to be processed.

Chapter 3 first discusses how to deal with confidentiality policies containing an arbitrary number of policy elements and thereby overcomes a first simplification assumption of the basic weakening approach developed in Chapter 2. Moreover, a further simplification assumption requiring all policy elements to be ground atoms is relaxed by allowing the usage of policy elements in the more expressive form of existentially quantified atoms. This leads to the construction of an approach that allows to vary the achieved level of confidentiality with the length of the employed weakening disjunctions and returns an inference-proof weakened view on an original database instance, whose knowledge does provably *not* enable an adversary to compromise a considered confidentiality policy. But this approach is still generic in the sense that a so-called clustering of the elements of a confidentiality policy to weakening disjunction templates, which are both credible in terms of confidentiality and meaningful in terms of availability with respect to a considered application scenario, is only discussed on the declarative level.

As a formal complexity analysis of this clustering problem indicates that weakening disjunction templates of a length longer than 2 can *not* be constructed efficiently in general, Chapter 4 then develops a concrete algorithmic instantiation of the clustering of policy elements to disjunction templates of length 2, which lead to weakenings by the shortest possible non-trivial disjunctions and thus maximize availability. This clustering is based on well-known and efficient algorithms for the computation of maximum matchings on (general) graphs modeling each potential secret to be clustered as a vertex and each admissible disjunction template of length 2 as an edge. Moreover, a concrete example of an (again availability-maximizing) notion of admissible indistinguishabilities specifying which policy elements might be possibly clustered to an admissible weakening disjunction is presented. In general, this notion is deliberately left generic to keep the weakening approach applicable for different application scenarios in the sense that the structure weakening disjunctions should have can be adapted.

To next overcome the simplification assumption that an adversary is *not* supposed to have any further a priori knowledge, Chapter 5 extends the availability-maximizing and efficient implementation of the generic weakening approach provided in Chapter 4 to be also confidentiality preserving under scenarios, in which an adversary has some a priori knowledge. This a priori knowledge is supposed to be expressible within a suitably restricted subclass of so-called Tuple Generating Dependencies, which are well-known in the field of relational databases. Under this extended setup an adversary's enhanced possibilities to compromise a confidentiality policy by employing his a priori knowledge are analyzed with scrutiny and suitable counter measures are developed to disable these additional inference-channels. This finally leads to an extended weakening approach, which provably preserves confidentiality in the sense of Controlled Interaction Execution under the considered subclass of a priori knowledge, but still remains computationally efficient even for large input instances.

Chapter 6 confirms this efficiency of the (extended) weakening approach experimentally with the help of a prototype implementation under different experiment setups and input instances in the order of magnitude of

- database instances with 1 000 000 tuples,
- confidentiality policies with up to 100 000 potential secrets and
- an adversary's a priori knowledge with up to 2500 dependencies.

For some restricted application scenarios, in which the matching computation turns out to have by far the greatest impact on the overall runtime, a slightly modified weakening algorithm employing a heuristic for the construction of (almost) maximum matchings to determine a clustering of the policy elements is presented. This modification leads to a considerable speedup and results only in a slight loss of availability, as also evaluated in this chapter.

7.2 Directions for Future Work

Although the generic weakening approach developed in Chapter 3 is able to employ disjunctions of an arbitrary length ≥ 2 to weaken confidential knowledge, a concrete implementation of its subroutine for the computation of extended clusterings (inducing weakening disjunction templates) is only specified for the construction of clusters of size 2. Hence, the development of concrete algorithms grouping the elements of a given confidentiality policy to (extended) clusters of a (minimum) size

of $k^* \geq 3$ – thereby still obeying a considered notion of admissible indistinguishabilities and minimizing the usage of artificial additional potential secrets – would obviously be a first useful extension of this thesis.

According to Section 3.4 this clustering problem is NP-hard in general and can hence *not* be solved efficiently in polynomial time in the size of its inputs as long as NP $\neq$ P is supposed to hold. As an immediate consequence, one might try to find reasonable heuristic solutions to this clustering problem by relaxing the optimization goal of minimizing the number of additional potential secrets to the less strict requirement that the number of actually employed additional potential secrets should not be "too far away" from an optimum solution. Additionally, one might also consider restricted subclasses of the extended clustering problem, for which reasonable heuristic solutions are easier to find or for which even optimum solutions can be found "efficiently enough".

Such a restricted subclass might for instance be the extended clustering problem, whose notion of admissible indistinguishabilities is instantiated with the interchangeability criterion as proposed in Section 4.2.1. According to this section, interchangeability – initially developed in Definition 4.5 for the construction of admissible clusters of size 2 – can naturally be extended to be also applicable for the construction of admissible clusters of a size larger than 2 and then results in indistinguishability graphs, which can be decomposed efficiently into maximal cliques (cf. [84, 85]) and each (subset) of these cliques consists of pairwise interchangeable potential secrets all sharing the same single differing position. The remaining problem then is to decide to which clique each of the vertices occurring in more than one clique should be uniquely assigned to obtain a set of disjoint clusters, for whose extension only a minimum number of additional potential secrets is needed. Although this problem seems to be NP-hard (cf. Section 4.2.1), too, it might nonetheless be possible to solve this problem exactly in a reasonable time, if most of the vertices of an indistinguishability graph are in only one clique.

After the development of algorithmic solutions to the construction of extended clusterings with clusters of a (minimum) size of $k^* \geq 3$, one might next try to analyze if and to what extent weakening disjunctions of a length ≥ 3 can guarantee the existence of a certain minimum number of "secure" alternative database instances for elements of a confidentiality policy, if an adversary is supposed to have a priori knowledge in the form of single premise tuple generating dependencies. For that purpose, one should first analyze, whether (and how) an adversary might employ this kind of a priori knowledge to reduce the number of credible alternative instances induced by sets of weakening disjunctions and should then, if necessary, develop counter measures to disable these inference-channels. In a next step, one could afterwards try to adapt Algorithm 5.1 – designed for handling

only disjunction templates of length 2 – to guarantee the existence of a certain minimum number of "secure" alternative instances and prove this property in the spirit of Theorem 3.1.

With regard to an adversary's a priori knowledge, another challenging question for future work is to find further versatile classes of a priori knowledge, under which inference-proofness can be provably guaranteed with the help of (possibly suitably adapted versions of) the weakening approach proposed in this thesis. A first class of such extended a priori knowledge might consist of Tuple Generating Dependencies having premises in the form of multiple conjunctively connected atoms (cf. [1, 54]) instead of just one single atom. First of all, such an extension raises the question which subset of atoms of the premises of a set of these dependencies (all interfering with a confidentiality policy) should be distorted, as the (non-)satisfaction of one of these atoms might influence the satisfaction-status of several premises of these dependencies – due to implication relationships between the atoms of the premises of (possibly different) dependencies. One should hence try to find a minimum subset of these atoms, whose distortion allows for the construction of (a sufficient number of) "secure" alternative database instances to establish inference-proofness, to keep the number of distorted atoms of premises as low as possible in terms of availability.

Moreover, further extensions of an adversary's a priori knowledge might consider other classes of a priori knowledge, such as the well-known classes of Equality Generating Dependencies [1, 53] or Inclusion Dependencies [1, 60]. While the former class of dependencies essentially expresses that certain components (in the form of constants assigned to corresponding attributes) of certain database tuples should be equal, the latter class can be seen as a special case of tuple generating dependencies requiring that the projection of a relational database instance r over a database schema $\langle R|\mathcal{A}_R|SC_R\rangle$ on a certain subset of attributes of $\mathcal{A}_R$ is completely contained in a corresponding projection of a relational database instance s over a (possibly different) database schema $\langle S|\mathcal{A}_S|SC_S\rangle$. As an immediate consequence, the introduction of inclusion dependencies would require to consider database instances with multiple relations – in contrast to the simplifying assumption met in Section 2.1 that a database instance with only a single relation is considered in this thesis. In general, for each further class of a priori knowledge an adversary's enhanced possibilities to compromise a confidentiality policy need to be analyzed with scrutiny in the spirit of Section 5.2 and then, if necessary, suitable counter measures need to be developed to disable these additional inference-channels.

In terms of availability knowledge should usually only be distorted, if an adversary might otherwise be able to compromise confidentiality requirements. But in this thesis, both Theorem 3.1 and Theorem 5.2 only prove that the developed

weakening algorithms suitably distort confidential knowledge in a sufficient way. While the generic weakening algorithm and its availability-maximizing instantiation do *not* seem to provide room for improvements of availability – at least, if their clustering stages remain instance-independent – Algorithm 5.1, which can also handle an adversary's a priori knowledge in the form of single premise tuple generating dependencies, seems to generally overestimate the threats posed by possible inference-channels. It might hence be worthwhile to analyze if and to what extend these threats are overestimated and how this overestimation might be reduced without losing the computational efficiency of this algorithm. Of course, the most desirable result of this analysis would be an improved weakening algorithm only introducing necessary distortions together with formal proofs that this improved algorithm suitably distorts confidential knowledge in a both sufficient and necessary way.

In Section 1.3.1 the enforcement of confidentiality requirements with the help of weakening disjunctions has been motivated by the well-known approaches of k-anonymization and ℓ-diversification [47, 70, 79, 86]. Thereby, k-anonymization aims at preventing the re-identification of individuals on the basis of so-called quasi-identifiers, which describe some of the individuals' properties, by generalizing these quasi-identifiers to such an extent that an individual can *not* be distinguished from $k-1$ other individuals on the basis of these quasi-identifiers.

Similarly, a weakening disjunction of length k should *not* enable an adversary to distinguish whether a certain potential secret of this disjunction or one of the $k-1$ other potential secrets of this disjunction is satisfied by the original instance. One could hence try to model k-anonymization within the weakening approach developed in this thesis by ensuring that this approach forms disjunctions suitably distorting the knowledge to which of k different individuals each quasi-identifier of a set of k corresponding quasi-identifiers, which are to be made pairwise indistinguishable, is allocated. This might for instance be achieved by developing a notion of admissible indistinguishabilities leading to disjunctions of the above mentioned structure, which are created over a confidentiality policy containing all potential secrets occurring as disjuncts within these disjunctions.

But even under k-anonymization confidentiality might still be breached, if all quasi-identifiers of a group of k pairwise indistinguishable quasi-identifiers are related to the same sensitive value: although an adversary can still *not* disclose which of the k pairwise indistinguishable quasi-identifiers is related to which of the k individuals represented by these quasi-identifiers (due to k-anonymization), he can nonetheless conclude that all of these individuals are definitely related to the considered sensitive value. To mitigate this inference-channel, ℓ-diversification

additionally requires that there should be at least ℓ pairwise different sensitive values, with which the quasi-identifiers of each group of k pairwise indistinguishable quasi-identifiers should be related. Within the weakening approach this additional requirement of ℓ-diversification imposes an additional restriction on the construction of weakening disjunctions, which might again be implemented by a corresponding notion of admissible indistinguishabilities.

Such a modeling of k-anonymization and ℓ-diversification within the weakening approach would have the great advantage that the property of inference-proofness in the sense of Controlled Interaction Execution, which is guaranteed for the weakening approach according to Theorem 3.1 and Theorem 5.2, would also hold for this modeling of k-anonymization and ℓ-diversification, while current standard approaches to k-anonymization and ℓ-diversification are usually *not* analyzed with respect to their achievements in confidentiality in a formal way. This is in particular true for scenarios, in which an adversary is supposed to have a kind of a priori knowledge, under which the weakening approach is known to be inference-proof.

Within this thesis the generation of weakened views is always analyzed under the (implicit) assumption that an adversary gets to know only one single weakened view on an original database instance. But in many real-world scenarios the data stored in such an instance might change over the time and there might hence be the wish to release an updated weakened view on a certain original database instance some time after a first weakened view on this instance has been released. Although the problem of updating data in an inference-proof way has generally been studied within the framework of Controlled Interaction Execution [20, 21, 31, 33], it is still an open problem in the context of inference-proof weakened views. Similar to the field of privacy preserving data publishing (cf. [56, 92]), the challenge is to ensure that an adversary knowing several weakened views on an original database instance – whose data have possibly changed in the course of time – can *not* take advantage of this knowledge to compromise a confidentiality policy.

From a practical point of view, another emerging question is how (extensions of commonly used) relational database management systems (cf. [1, 75, 78]) can operate on weakened views on relational database instances. Thereby, this question naturally comprises the problem of a suitable representation of disjunctive knowledge within relational database management systems as well as the challenge of finding suitable extensions of (the semantics of) commonly used operators of the relational algebra, which are able to handle disjunctive knowledge.

Bibliography

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, Reading, 1995.

[2] Gagan Aggarwal, Mayank Bawa, Prasanna Ganesan, Hector Garcia-Molina, Krishnaram Kenthapadi, Rajeev Motwani, Utkarsh Srivastava, Dilys Thomas, and Ying Xu. Two can keep a secret: A distributed architecture for secure database services. In *2nd Biennial Conference on Innovative Data Systems Research, CIDR 2005*, pages 186–199, 2005.

[3] Gagan Aggarwal, Tomás Feder, Krishnaram Kenthapadi, Rajeev Motwani, Rina Panigrahy, Dilys Thomas, and An Zhu. Anonymizing tables. In Thomas Eiter and Leonid Libkin, editors, *10th International Conference on Database Theory, ICDT 2005*, volume 3363 of *Lecture Notes in Computer Science*, pages 246–258, Heidelberg, 2005. Springer.

[4] Leopoldo E. Bertossi and Lechen Li. Achieving data privacy through secrecy views and null-based virtual updates. *IEEE Transactions on Knowledge and Data Engineering*, 25(5):987–1000, 2013.

[5] Matt Bishop. *Computer Security: Art and Science*. Addison-Wesley, Boston, MA, 2003.

[6] Joachim Biskup. For unknown secrecies refusal is better than lying. *Data & Knowledge Engineering*, 33(1):1–23, 2000.

[7] Joachim Biskup. *Security in Computing Systems – Challenges, Approaches and Solutions*. Springer, Heidelberg, 2009.

[8] Joachim Biskup. Usability confinement of server reactions: Maintaining inference-proof client views by Controlled Interaction Execution. In Shinji Kikuchi, Shelly Sachdeva, and Subhash Bhalla, editors, *Databases in Networked Information Systems, DNIS 2010*, volume 5999 of *Lecture Notes in Computer Science*, pages 80–106, Heidelberg, 2010. Springer.

[9] Joachim Biskup. Inference control. In Henk C.A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security*, pages 600–605. Springer, New York, NY, 2nd edition, 2011.

[10] Joachim Biskup. Inference-usability confinement by maintaining inference-proof views of an information system. *International Journal of Computational Science and Engineering*, 7(1):17–37, 2012.

[11] Joachim Biskup. Logic-oriented confidentiality policies for controlled interaction execution. In Aastha Madaan, Shinji Kikuchi, and Subhash Bhalla, editors, *Databases in Networked Information Systems, DNIS 2013*, volume 7813 of *Lecture Notes in Computer Science*, pages 1–22, Heidelberg, 2013. Springer.

[12] Joachim Biskup. Selected results and related issues of confidentiality-preserving controlled interaction execution. In Marc Gyssens and Guillermo Ricardo Simari, editors, *9th International Symposium on Foundations of Information and Knowledge Systems, FoIKS 2016*, volume 9616 of *Lecture Notes in Computer Science*, pages 211–234, Cham, 2016. Springer International Publishing Switzerland.

[13] Joachim Biskup and Piero A. Bonatti. Controlled query evaluation for enforcing confidentiality in complete information systems. *International Journal of Information Security*, 3(1):14–27, 2004.

[14] Joachim Biskup and Piero A. Bonatti. Controlled query evaluation for known policies by combining lying and refusal. *Annals of Mathematics and Artificial Intelligence*, 40(1–2):37–62, 2004.

[15] Joachim Biskup and Piero A. Bonatti. Controlled query evaluation with open queries for a decidable relational submodel. *Annals of Mathematics and Artificial Intelligence*, 50(1–2):39–77, 2007.

[16] Joachim Biskup, Piero A. Bonatti, Clemente Galdi, and Luigi Sauro. Optimality and complexity of inference-proof data filtering and CQE. In Miroslaw

Kutylowski and Jaideep Vaidya, editors, *19th European Symposium on Research in Computer Security, ESORICS 2014*, volume 8713 of *Lecture Notes in Computer Science*, pages 165–181, Cham, 2014. Springer International Publishing Switzerland.

[17] Joachim Biskup, Martin Bring, and Michael Bulinski. Confidentiality preserving evaluation of open relational queries. In Tadeusz Morzy, Patrick Valduriez, and Ladjel Bellatreche, editors, *19th East European Conference on Advances in Databases and Information Systems, ADBIS 2015*, volume 9282 of *Lecture Notes in Computer Science*, pages 431–445, Cham, 2015. Springer International Publishing Switzerland.

[18] Joachim Biskup, Christine Dahn, Katharina Diekmann, Ralf Menzel, Dirk Schalge, and Lena Wiese. Publishing inference-proof relational data: An implementation and experiments. Submitted for publication, 2015.

[19] Joachim Biskup, David W. Embley, and Jan-Hendrik Lochner. Reducing inference control to access control for normalized database schemas. *Information Processing Letters*, 106(1):8–12, 2008.

[20] Joachim Biskup, Christian Gogolin, Jens Seiler, and Torben Weibert. Requirements and protocols for inference-proof interactions in information systems. In Michael Backes and Peng Ning, editors, *14th European Symposium on Research in Computer Security, ESORICS 2009*, volume 5789 of *Lecture Notes in Computer Science*, pages 285–302, Heidelberg, 2009. Springer.

[21] Joachim Biskup, Christian Gogolin, Jens Seiler, and Torben Weibert. Inference-proof view update transactions with forwarded refreshments. *Journal of Computer Security*, 19(3):487–529, 2011.

[22] Joachim Biskup, Sven Hartmann, Sebastian Link, and Jan-Hendrik Lochner. Chasing after secrets in relational databases. In Alberto H. F. Laender and Laks V. S. Lakshmanan, editors, *Proceedings of the 4th Alberto Mendelzon International Workshop on Foundations of Data Management, AMW 2010*, volume 619 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

[23] Joachim Biskup, Sven Hartmann, Sebastian Link, and Jan-Hendrik Lochner. Efficient inference control for open relational queries. In Sara Foresti and Sushil Jajodia, editors, *Data and Applications Security and Privacy XXIV – 24th Annual IFIP WG 11.3 Working Conference, DBSec 2010*, volume 6166 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 2010.

[24] Joachim Biskup, Sven Hartmann, Sebastian Link, Jan-Hendrik Lochner, and Torsten Schlotmann. Signature-based inference-usability confinement for relational databases under functional and join dependencies. In Nora Cuppens-Boulahia, Frédéric Cuppens, and Joaquín García-Alfaro, editors, *Data and Applications Security and Privacy XXVI, DBSec 2012*, volume 7371 of *Lecture Notes in Computer Science*, pages 56–73, Heidelberg, 2012. Springer.

[25] Joachim Biskup and Jan-Hendrik Lochner. Enforcing confidentiality in relational databases by reducing inference control to access control. In Juan A. Garay, Arjen K. Lenstra, Masahiro Mambo, and René Peralta, editors, *10th International Conference on Information Security, ISC 2007*, volume 4779 of *Lecture Notes in Computer Science*, pages 407–422. Springer, 2007.

[26] Joachim Biskup, Jan-Hendrik Lochner, and Sebastian Sonntag. Optimization of the controlled evaluation of closed relational queries. In Dimitris Gritzalis and Javier Lopez, editors, *Emerging Challenges for Security, Privacy and Trust, IFIP 2009*, volume 297 of *Advances in Information and Communication Technology*, pages 214–225. Springer, 2009.

[27] Joachim Biskup and Marcel Preuß. Database fragmentation with encryption: Under which semantic constraints and a priori knowledge can two keep a secret? In Lingyu Wang and Basit Shafiq, editors, *Data and Applications Security and Privacy XXVII – 27th Annual IFIP WG 11.3 Conference, DBSec 2013*, volume 7964 of *Lecture Notes in Computer Science*, pages 17–32, Heidelberg, 2013. Springer.

[28] Joachim Biskup and Marcel Preuß. Inference-proof data publishing by minimally weakening a database instance. In Atul Prakash and Rudrapatna Shyamasundar, editors, *Information Systems Security – 10th International Conference, ICISS 2014*, volume 8880 of *Lecture Notes in Computer Science*, pages 30–49, Cham, 2014. Springer International Publishing Switzerland.

[29] Joachim Biskup and Marcel Preuß. Information control by policy-based relational weakening templates. In Ioannis Askoxylakis, Sotiris Ioannidis, Sokratis Katsikas, and Catherine Meadows, editors, *21st European Symposium on Research in Computer Security, ESORICS 2016, Part II*, volume 9879 of *Lecture Notes in Computer Science*, pages 1–21, Cham, 2016. Springer International Publishing Switzerland. To appear.

[30] Joachim Biskup, Marcel Preuß, and Lena Wiese. On the inference-proofness of database fragmentation satisfying confidentiality constraints. In Xuejia Lai, Jianying Zhou, and Hui Li, editors, *14th Information Security Conference, ISC 2011*, volume 7001 of *Lecture Notes in Computer Science*, pages 246–261, Heidelberg, 2011. Springer.

[31] Joachim Biskup, Jens Seiler, and Torben Weibert. Controlled query evaluation and inference-free view updates. In Ehud Gudes and Jaideep Vaidya, editors, *Data and Applications Security XXIII, DBSec 2009*, volume 5645 of *Lecture Notes in Computer Science*, pages 1–16, Heidelberg, 2009. Springer.

[32] Joachim Biskup and Cornelia Tadros. Policy-based secrecy in the runs & systems framework and controlled query evaluation. In Isao Echizen, Noboru Kunihiro, and Ryôichi Sasaki, editors, *Advances in Information and Computer Security - 5th International Workshop on Security, IWSEC 2010*, pages 60–77. Information Processing Society of Japan (IPSJ), 2010.

[33] Joachim Biskup and Cornelia Tadros. Inference-proof view update transactions with minimal refusals. In Joaquín García-Alfaro, Guillermo Navarro-Arribas, Nora Cuppens-Boulahia, and Sabrina De Capitani di Vimercati, editors, *Data Privacy Management and Autonomous Spontaneous Security - 6th International Workshop, DPM 2011*, volume 7122 of *Lecture Notes in Computer Science*, pages 104–121. Springer, 2011.

[34] Joachim Biskup and Cornelia Tadros. Preserving confidentiality while reacting on iterated queries and belief revisions. *Annals of Mathematics and Artificial Intelligence*, 73(1-2):75–123, 2015.

[35] Joachim Biskup and Torben Weibert. Keeping secrets in incomplete databases. *International Journal of Information Security*, 7(3):199–217, 2008.

[36] Joachim Biskup and Lena Wiese. Preprocessing for controlled query evaluation with availability policy. *Journal of Computer Security*, 16(4):477–494, 2008.

[37] Joachim Biskup and Lena Wiese. Combining consistency and confidentiality requirements in first-order databases. In Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio Agostino Ardagna, editors, *12th International Conference on Information Security, ISC 2009*, volume 5735 of *Lecture Notes in Computer Science*, pages 121–134. Springer, 2009.

[38] Joachim Biskup and Lena Wiese. A sound and complete model-generation procedure for consistent and confidentiality-preserving databases. *Theoretical Computer Science*, 412(31):4044–4072, 2011.

[39] Jeremiah Blocki and Ryan Williams. Resolving the complexity of some data privacy problems. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *37th International Colloquium on Automata, Languages and Programming, ICALP 2010, Part II*, volume 6199 of *Lecture Notes in Computer Science*, pages 393–404, Heidelberg, 2010. Springer.

[40] Piero A. Bonatti, Sarit Kraus, and V. S. Subrahmanian. Foundations of secure deductive databases. *IEEE Transactions on Knowledge and Data Engineering*, 7(3):406–422, 1995.

[41] Boost Graph Library. Maximum cardinality matching, 2016. `http://www.boost.org/doc/libs/1_54_0/libs/graph/doc/maximum_matching.html`.

[42] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Universitext. Springer, Berlin, 2nd edition, 2001.

[43] Alina Campan and Traian Marius Truta. Data and structural k-anonymity in social networks. In Francesco Bonchi, Elena Ferrari, Wei Jiang, and Bradley Malin, editors, *Privacy, Security, and Trust in KDD, PinKDD 2008*, volume 5456 of *Lecture Notes in Computer Science*, pages 33–54, Heidelberg, 2008. Springer.

[44] Silvana Castano, Maria Grazia Fugini, Giancarlo Martella, and Pierangela Samarati. *Database Security*. ACM Press Books. Addison-Wesley, Wokingham, 1995.

[45] Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Enforcing confidentiality constraints on sensitive databases with lightweight trusted clients. In Ehud Gudes and Jaideep Vaidya, editors, *Data and Applications Security XXIII, DBSec 2009*, volume 5645 of *Lecture Notes in Computer Science*, pages 225–239, Heidelberg, 2009. Springer.

[46] Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Keep a few: Outsourcing data while maintaining confidentiality. In Michael Backes and Peng Ning, editors, *14th European Symposium on Research in Computer Security, ESORICS 2009*, volume 5789 of *Lecture Notes in Computer Science*, pages 440–455, Heidelberg, 2009. Springer.

[47] Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, and Pierangela Samarati. k-Anonymity. In Ting Yu and Sushil Jajodia, editors, *Secure Data Management in Decentralized Systems*, volume 33 of *Advances in Information Security*, pages 323–353. Springer, New York, NY, 2007.

[48] Stephen Cook. The P versus NP Problem. In James Carlson, Arthur Jaffe, and Andrew Wiles, editors, *The Millennium Prize Problems*, pages 87–104. American Mathematical Society, Providence, RI, 2006. Published for the Clay Mathematics Institute, Cambridge, MA.

[49] Dorothy E. Denning. *Cryptography and Data Security*. Addison-Wesley, Reading, 1982.

[50] Reinhard Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer, Heidelberg, 4th edition, 2012.

[51] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. A Wiley-Interscience Publication. John Wiley & Sons, New York, NY, 2nd edition, 2001.

[52] Marcel Erné. *Einführung in die Ordnungstheorie*. Bibliographisches Institut, Zürich, 1982.

[53] Ronald Fagin. Equality-Generating Dependencies. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 1009–1010. Springer, New York, NY, 2009.

[54] Ronald Fagin. Tuple-Generating Dependencies. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 3201–3202. Springer, New York, NY, 2009.

[55] Csilla Farkas and Sushil Jajodia. The inference problem: A survey. *ACM SIGKDD Explorations Newsletter*, 4(2):6–11, 2002.

[56] Benjamin C.M. Fung, Ke Wang, Ada Wai-Chee Fu, and Philip S. Yu. *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. Data Mining and Knowledge Discovery. CRC Press, Boca Raton, FL, 2011.

[57] Vignesh Ganapathy, Dilys Thomas, Tomás Feder, Hector Garcia-Molina, and Rajeev Motwani. Distributing data for secure database services. *Transactions on Data Privacy*, 5(1):253–272, 2012.

[58] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, CA, 1979.

[59] Dieter Gollmann. *Computer Security*. John Wiley and Sons, Chichester, 2nd edition, 2006.

[60] Marc Gyssens. Database Dependencies. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 704–708. Springer, New York, NY, 2009.

[61] Joseph Y. Halpern and Kevin R. O'Neill. Secrecy in multiagent systems. *ACM Transactions on Information and System Security*, 12(1), 2008.

[62] Michael Hay, Gerome Miklau, David Jensen, Donald F. Towsley, and Chao Li. Resisting structural re-identification in anonymized social networks. *VLDB Journal*, 19(6):797–823, 2010.

[63] Maurice Herlihy and Nir Shavit. *The Art of Multiprocessor Programming*. Morgan Kaufmann, Amsterdam, 2008.

[64] Richard M. Karp and Michael Sipser. Maximum matchings in sparse random graphs. In *Symposium on Foundations of Computer Science, FOCS 1981*, pages 364–375. IEEE Computer Society, 1981.

[65] Jon Kleinberg and Éva Tardos. *Algorithm Design*. Pearson Education, Boston, MA, 2006.

[66] Donald E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. ACM Press, New York, NY, 1993.

[67] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Algorithms and Combinatorics. Springer, Heidelberg, 5th edition, 2012.

[68] Hector J. Levesque and Gerhard Lakemeyer. *The Logic of Knowledge Bases*. The MIT Press, Cambridge, MA, 2000.

[69] Jan-Hendrik Lochner. *An Effective and Efficient Inference Control System for Relational Database Queries*. PhD thesis, Dortmund University of Technology, 2011.

[70] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1), 2007.

[71] Jakob Magun. Greedy matching algorithms: An experimental study. *ACM Journal of Experimental Algorithmics*, 3(6), 1998.

[72] Kurt Mehlhorn and Stefan Näher. *LEDA: A platform for combinatorial and geometric computing*. Cambridge University Press, Cambridge, 1999.

[73] Silvio Micali and Vijay V. Vazirani. An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs. In *Symposium on Foundations of Computer Science, FOCS 1980*, pages 17–27, 1980.

[74] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, Boston, MA, 1997.

[75] Shamkant B. Navathe and Ramez Elmasri. *Fundamentals of Database Systems*. Pearson, Boston, MA, 7th edition, 2016.

[76] Anil Nerode and Richard A. Shore. *Logic for Applications.* Graduate Texts in Computer Science. Springer, New York, NY, 2nd edition, 1997.

[77] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications, Mineola, NY, 1998.

[78] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems.* McGraw-Hill, Boston, MA, 3rd edition, 2003.

[79] Pierangela Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[80] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency.* Algorithms and Combinatorics. Springer, Berlin, 2003.

[81] Robin Sibson and Nicholas Jardine. *Mathematical Taxonomy.* Wiley series in probability and mathematical statistics. Wiley, London, 1971.

[82] George L. Sicherman, Wiebren de Jonge, and Reind P. van de Riet. Answering queries without revealing secrets. *ACM Transactions on Database Systems*, 8(1):41–59, 1983.

[83] Günther Stiege. Playing with Knuth's words.dat. Technical Report 1/12, Department of Computer Science, University of Oldenburg, Germany, May 2012.

[84] Günther Stiege. Cliques in Graphs of Type WORDS. Technical Report 2/13, Department of Computer Science, University of Oldenburg, Germany, June 2013.

[85] Günther Stiege. Knuth Graphs. Technical Report 2/15, Department of Computer Science, University of Oldenburg, Germany, December 2015.

[86] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[87] Andrew S. Tanenbaum and Herbert Bos. *Modern Operating Systems.* Pearson, Boston, MA, 4th edition, 2015.

[88] Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter. *Handbook of Knowledge Representation.* Foundations of Artificial Intelligence. Elsevier, Amsterdam, 2008.

[89] Vijay V. Vazirani. 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.

[90] Ingo Wegener. *Complexity Theory: Exploring the Limits of Efficient Algorithms*. Springer, Berlin, 2005.

[91] Lena Wiese. *Preprocessing for Controlled Query Evaluation in Complete First-Order Databases*. PhD thesis, Dortmund University of Technology, 2009.

[92] Raymond Chi-Wing Wong and Ada Wai-Chee Fu. *Privacy-Preserving Data Publishing – An Overview*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, San Rafael, CA, 2010.