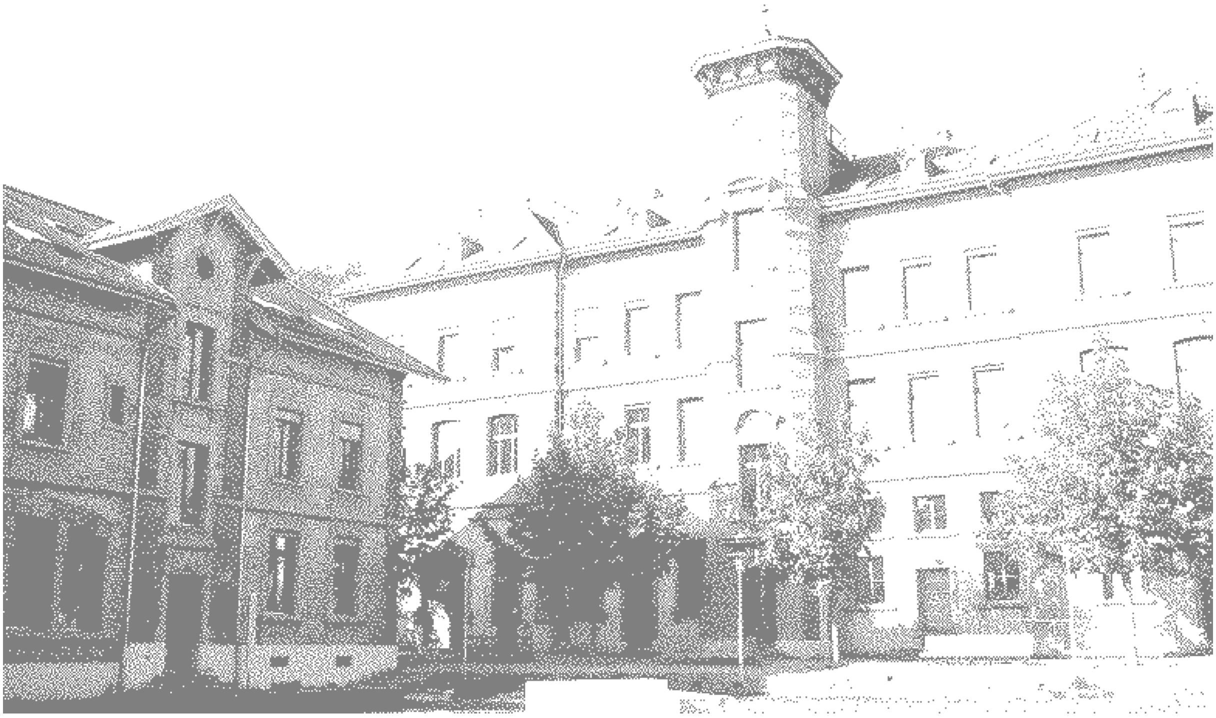
b
UNIVERSITÄT
BERN

Data Privacy for $\mathcal{ALC}$ Knowledge Bases

Ph. Stouppa, Th. Studer

Technischer Bericht IAM-08-002 vom 29. August 2008

Institut für Informatik und angewandte Mathematik, www.iam.unibe.ch



Data Privacy for $\mathcal{ALC}$ Knowledge Bases

Phiniki Stouppa, Thomas Studer

Technischer Bericht IAM-08-002 vom 29. August 2008

CR Categories and Subject Descriptors:
I.2.4 [Knowledge Representation Formalisms and Methods]: Miscellaneous; H.2.7 [Database Administration]: Security, integrity, and protection; F.4.1 [Mathematical Logic]: Proof theory

General Terms:
Theory, security

Additional Key Words:
Data privacy, description logic

Institut für Informatik und angewandte Mathematik, Universität Bern

Abstract

Information systems support data privacy by granting access only to certain (public) views. The data privacy problem is to decide whether hidden (private) information may be inferred from the public views and some additional general background knowledge. We study this problem in the context of $\mathcal{ALC}$ knowledge bases. First we show that the $\mathcal{ALC}$ privacy problem wrt. concept retrieval and subsumption queries is ExpTime-complete. Then we provide a sufficient (but not necessary) condition for data privacy that can be checked in PTime. This second approach is directly applicable to modular ontologies.

Contents

1 Introduction

In information systems, the problem of data privacy is to verify whether the *confidential* information that is stored in a system is not provided to unauthorized users and therefore, personal and other sensitive data remain private. Data privacy issues are particularly critical in environments where sharing and reuse of information are constantly applied. Let us cite the OWL Language Guide [1]: '...the capability to merge data from multiple sources, combined with the inferential power of OWL, does have potential for abuse. Users of OWL should be alert to the potential privacy implications.'

A definition of data privacy that applies both on database and knowledge base systems has been presented in [2, 3]. There, the problem of *provable data privacy on views* was introduced as follows. Assume that all information about a system is provided to a user through a view and some background knowledge that is publicly available. The privacy problem under this setting is to decide that the user cannot infer - from the view and the background knowledge - any answer to a given query q. That one cannot infer any answer to q is formalized as *the set of certain answers to* q *is empty*. If the problem is answered positively, we say that privacy is *preserved* for q. The notion of certain answer originates from the study of incomplete databases [4] and is now a key notion in data integration [5, 6, 7] and data exchange [8, 9]. Provable data privacy was studied in [2] from the perspective of relational databases. We then extended the notion of provable data privacy in order to make it applicable also to knowledge base systems, see [3]. There we showed that this privacy problem on a view reduces to the retrieval or entailment problem.

In the present paper, we will use the notion of provable privacy to study a more general problem: the problem of (deciding) *data privacy on view definitions*. The new problem is now the following: given only a view definition instead of a complete view, decide whether privacy is preserved on all possible views of that view definition. We investigate the new problem for the case of $\mathcal{ALC}$ knowledge bases with GCIs. In such a knowledge base the domain is only partially known (incomplete), background knowledge is formalized as a part of the knowledge base, and for the view and the privacy condition we allow for concept retrieval and subsumption queries.

We present a total and a partial solution to the problem. The total solution shows that the problem is decidable and can be computed by considering only a finite number of possible views. As a corollary we obtain that the problem is ExpTime-complete.

The partial solution detects only some of the privacy preserving cases and is based on a syntactic criterion on the view definition, the general knowledge base and the privacy condition. More specifically, it makes use of the observation that quantifiers restrict the amount of information one can access through a concept. For example, given that all R-assertions are hidden, the validity of the concept $\forall R.A$ does not reveal any information (i.e. validity or specific individuals) about the concept A. We show that this syntactic condition is sufficient (but not necessary) for data privacy and that it can be checked in PTime. Moreover, this criterion may be also applied to certain modular and $\mathcal{E}$-connected $\mathcal{ALC}$ ontologies [10, 11].[1] Consider for instance the scenario where a public ontology $\mathcal{O}_1$ consists of concepts in which concepts of a second, distinct ontology $\mathcal{O}_2$ might appear only behind some quantified roles E_i.[2] If there is no public role assertion $(x, y) : E_i$ for any E_i that is universally bound in $\mathcal{O}_1$, then privacy is preserved for all (non-trivial) concepts of $\mathcal{O}_2$ under any view of $\mathcal{O}_1$.

Our notion of privacy is based on the concept of certain answers. Another very important privacy notion is that of *perfect privacy* which assures that no information at all about the answers to a given query is exhibited by the public data. If, for example, a view reveals something about the number of answers to q, then q is considered insecure. Perfect privacy has been introduced in [12] and generalized in [13]. Recently, a connection between perfect privacy and query containment has been established [14] which allows to identify subclasses of conjunctive queries for which enforcing perfect privacy is tractable.

An active research area in the field of data privacy is also the development and evaluation of privacy preserving query answering methodologies for relational databases. The question of how much information a view reveals and whether it leaks private data is addressed in [15] for a variety of confidentiality policies. There it is argued that, in overcoming a privacy violation, a refusal (i.e. deny to answer a query) should be in general preferable rather than a lie (i.e. give a false answer). In [16] a generalized answer is proposed whereas in [17, 18] it is proposed to return a partial answer. In the latter case, the problem consists in inferring a maximal subset of the answer to a query so that no secrets are violated.

The rest of the paper is organized as follows: we first present the syntax and the semantics of the language $\mathcal{ALC}$, explain how a query is answered on an $\mathcal{ALC}$ knowledge base and introduce the problem of data privacy on

[1] This can be obtained by appropriately adapting the definition of the privacy condition as well as the tableaux methods used in proving the results.

[2] Note that role inverses are ruled out.

a view. Then, in Section 3 we define data privacy on a view definition and show both the complete and partial solutions to the problem. The complexities of the problems are also discussed here. In Section 4, we continue with a deductive system for $\mathcal{ALC}$ which will be used in the proofs of the two solutions; these are presented in Section 5. Finally, in Section 6 we summarize our results and give some directions for further work.

2 Preliminaries

The language of $\mathcal{ALC}$ consists of a countable set of *individuals* Ind, a countable set of *atomic concepts* AConc, a countable set of *roles* Rol and the *concepts* built on AConc and Rol as follows:

$$C, D := A \mid \neg A \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$$

where $A \in \mathsf{AConc}$, $R \in \mathsf{Rol}$, and C and D are concepts. We use $a, b, c, \ldots$ to denote individuals, $A, A_1, A_2, \ldots$ to denote atomic concepts, $R, R_1, R_2, \ldots$ to denote roles and $C, C_1, C_2, \ldots, D, D_1, D_2, \ldots$ to denote concepts.
Note that the language includes only concepts in negation normal form. The complement of a concept $\neg(C)$ is inductively defined, as usual, by using the law of double negation, de Morgan's laws and the dualities for quantifiers. When the scope of the negation is unambiguous, we also write $\neg C$ instead of $\neg(C)$. Moreover, the constants $\top$ and $\bot$ abbreviate $A \sqcup \neg A$ and $A \sqcap \neg A$, respectively, for some $A \in \mathsf{AConc}$.
We also introduce the (non-standard) notion of subterms of a given concept. The set of *subterms* $s(C)$ of a concept C is inductively defined by:

$$\begin{array}{rlrl} s(A) := & \{A\} & s(\neg A) := & \{\neg A\} \\ s(C \star D) := & \{C \star D\} \cup s(C) \cup s(D) & s(QR.C) := & \{QR.C\} \cup s(C) \end{array}$$

where $\star$ is either $\sqcup$ or $\sqcap$ and Q is either $\forall$ or $\exists$. Note that the complements of atomic concepts are not decomposable. For instance, the subterms of $A_1 \sqcup \exists R.\neg A_2$ are $A_1, \neg A_2, \exists R.\neg A_2$ and $A_1 \sqcup \exists R.\neg A_2$.

Concepts are interpreted in the usual way:

Definition An interpretation $\mathcal{I}$ consists of a non-empty domain $\Delta^{\mathcal{I}}$ and a mapping $()^{\mathcal{I}}$ that assigns

- to each individual $a \in \mathsf{Ind}$ an element $(a)^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
- to each atomic concept $A \in \mathsf{AConc}$ a set $(A)^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
- to each role $R \in \mathsf{Rol}$ a relation $(R)^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

The elements of a domain are denoted by $d, d_1, d_2, \ldots$. The interpretation $\mathcal{I}$ extends then on concepts as follows:

$$\begin{array}{rcl} (\neg A)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus (A)^{\mathcal{I}} \\ (C \sqcap D)^{\mathcal{I}} & = & (C)^{\mathcal{I}} \cap (D)^{\mathcal{I}} \\ (C \sqcup D)^{\mathcal{I}} & = & (C)^{\mathcal{I}} \cup (D)^{\mathcal{I}} \\ (\forall R.C)^{\mathcal{I}} & = & \{d_1 \in \Delta^{\mathcal{I}} \mid \forall d_2\ ((d_1, d_2) \in (R)^{\mathcal{I}} \Rightarrow d_2 \in (C)^{\mathcal{I}})\} \\ (\exists R.C)^{\mathcal{I}} & = & \{d_1 \in \Delta^{\mathcal{I}} \mid \exists d_2\ ((d_1, d_2) \in (R)^{\mathcal{I}}\ \&\ d_2 \in (C)^{\mathcal{I}})\} \end{array}$$

We can now define the notion of a knowledge base and its models. An $\mathcal{ALC}$ knowledge base $\mathcal{O}$ is the union of

1. a finite *terminological* set (TBox) of *inclusion axioms* that have the form $\top \sqsubseteq C$,[3] where C is called *inclusion concept*, and
2. a finite *assertional* set (ABox) of assertions of the form $a : C$ (*concept assertion*) or $(a, b) : R$ (*role assertion*) where R is called *assertional role* and C is called *assertional concept*.

We denote the set of individuals that appear in $\mathcal{O}$ by $\mathsf{Ind}(\mathcal{O})$. An interpretation $\mathcal{I}$ is a *model* of

- an inclusion axiom $\top \sqsubseteq C$ ($\mathcal{I} \models \top \sqsubseteq C$) if $(C)^{\mathcal{I}} = \Delta^{\mathcal{I}}$,
- a concept assertion $a : C$ ($\mathcal{I} \models a : C$) if $(a)^{\mathcal{I}} \in (C)^{\mathcal{I}}$,
- a role assertion $(a, b) : R$ ($\mathcal{I} \models (a, b) : R$) if $((a)^{\mathcal{I}}, (b)^{\mathcal{I}}) \in (R)^{\mathcal{I}}$.

Let $\mathcal{O}$ be the $\mathcal{ALC}$-knowledge base of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. An interpretation $\mathcal{I}$ is a model of $\mathcal{O}$ if $\mathcal{I} \models \phi$, for every $\phi \in \mathcal{T} \cup \mathcal{A}$. A knowledge base $\mathcal{O}$ is *consistent* if it has a model. Moreover, for ψ an inclusion axiom or an assertion, we say that $\mathcal{O} \models \psi$ (in words, $\mathcal{O}$ *entails* ψ) if for every model $\mathcal{I}$ of $\mathcal{O}$, $\mathcal{I} \models \psi$ also holds.
Deciding the consistency of an $\mathcal{ALC}$ knowledge base is an ExpTime-complete problem, see for instance [19][4]. The entailment problem is reducible to the consistency problem as follows:

Theorem 2.1. *Let $\mathcal{O}$ be a knowledge base and $n_{ew} \in \mathsf{Ind} \setminus \mathsf{Ind}(\mathcal{O})$. Then,*

- *$\mathcal{O} \models \top \sqsubseteq C$ iff $\mathcal{O} \cup \{n_{ew} : \neg C\}$ is inconsistent and*
- *$\mathcal{O} \models a : C$ iff $\mathcal{O} \cup \{a : \neg C\}$ is inconsistent.*

Theorem 2.1 shows that an entailment can be decided in ExpTime. Moreover, the inconsistency problem is reducible to the entailment problem and so, deciding an entailment is an ExpTime-complete problem, too.
The reasoning tasks on an $\mathcal{ALC}$ knowledge base are formulated below as *queries*. For the time being we consider only subsumption and retrieval queries.

[3] This form does not restrict a knowledge base since an arbitrary inclusion $C_1 \sqsubseteq C_2$ can be linearly transformed to its equivalent $\top \sqsubseteq \neg C_1 \sqcup C_2$.

[4] More details are available from the DL complexity navigator at `http://www.cs.man.ac.uk/~ezolin/dl/`.

Definition An $\mathcal{ALC}$ *query* q is either a concept of $\mathcal{ALC}$ (called retrieval query) or an inclusion axiom (called boolean query). The *answer to a query* q with respect to a knowledge base $\mathcal{O}$ ($\mathsf{ans}(q,\mathcal{O})$) is given as follows where tt is a special constant denoting 'true'.

$$\begin{array}{rcl} \mathsf{ans}(\top \sqsubseteq C, \mathcal{O}) & := & \{\mathsf{tt}\}\text{ , if } \mathcal{O} \models \top \sqsubseteq C, \\ \mathsf{ans}(\top \sqsubseteq C, \mathcal{O}) & := & \emptyset\text{ , if } \mathcal{O} \not\models \top \sqsubseteq C, \\ \mathsf{ans}(C, \mathcal{O}) & := & \{a \in \mathsf{Ind}(\mathcal{O}) \mid \mathcal{O} \models a : C\}\ . \end{array}$$

A *view definition* V is a finite set of $\mathcal{ALC}$ queries.

A view of a given view definition is a function that maps each query of the view definition to an answer. Formally, we define it as follows.

Definition A *view* V_I of a view definition V is a set of tuples $\langle q_i, r_i\rangle$ such that

1. for every $q \in V$ there exists r with $\langle q, r\rangle \in V_I$,
2. $\{\langle q, r\rangle, \langle q, r'\rangle\} \subseteq V_I$ implies $r = r'$, and
3. if $\langle q, r\rangle \in V_I$, then
   (a) $q \in V$,
   (b) $r \subseteq \mathsf{Ind}$ and finite if q is a retrieval query,
   (c) $r \subseteq \{\mathsf{tt}\}$ if q is a boolean query.

A knowledge base $\mathcal{O}$ entails a view V_I ($\mathcal{O} \models V_I$) if $r = ans(q, \mathcal{O})$, for each $\langle q, r\rangle \in V_I$.

We turn now to the problem of provable data privacy wrt. views. This problem has been examined for arbitrary data and knowledge bases in [2, 3]. Here we present the problem from the point of view of $\mathcal{ALC}$ knowledge bases and queries; we additionally admit that the underlying knowledge base is always consistent.
The problem assumes that a user is granted access to a specific view V_I and to some general (background) knowledge of such a knowledge base. In our case we assume that all information about the knowledge base is stated explicitly in it and, therefore, the background knowledge coincides with a part of the knowledge base. We call this knowledge base $\mathcal{O}_{bg}$. Note that in our setting $\mathcal{O}_{bg}$ is not reducible to V_I since role assertions are not expressible in views.

Informally, we say that *data privacy is preserved* for a query q with respect to $\langle \mathcal{O}_{bg}, V_I \rangle$ if there are no answers to q that follow with certainty from the information of V_I and $\mathcal{O}_{bg}$. This can be made precise by the notion of certain answer. The function $\mathsf{certain}(q, \langle \mathcal{O}_{bg}, V_I \rangle)$ returns the answers to q that hold in every knowledge base that - according to the user's knowledge - could be the actual one (a so-called *possible* knowledge base).

Definition A knowledge base $\mathcal{P}$ is *possible* with respect to $\langle \mathcal{O}_{bg}, V_I \rangle$ if $\mathcal{P}$ is consistent, $\mathcal{O}_{bg} \subseteq \mathcal{P}$, and $\mathcal{P} \models V_I$. By $\mathsf{Poss}_{\langle \mathcal{O}_{bg}, V_I \rangle}$, we denote the set of all possible knowledge bases with respect to $\langle \mathcal{O}_{bg}, V_I \rangle$.

In the sequel we consider only $\langle \mathcal{O}_{bg}, V_I \rangle$ tuples with $\mathsf{Poss}_{\langle \mathcal{O}_{bg}, V_I \rangle} \neq \emptyset$.

Definition The *certain answers* to a query q with respect to $\langle \mathcal{O}_{bg}, V_I \rangle$ are defined by

$$\mathsf{certain}(q, \langle \mathcal{O}_{bg}, V_I \rangle) := \bigcap_{\mathcal{P} \in \mathsf{Poss}_{\langle \mathcal{O}_{bg}, V_I \rangle}} \mathsf{ans}(q, \mathcal{P}).$$

Definition Given a knowledge base $\mathcal{O}_{bg}$, a view V_I and a query q, *data privacy is preserved for* q *with respect to* $\langle \mathcal{O}_{bg}, V_I \rangle$ if

$$\mathsf{certain}(q, \langle \mathcal{O}_{bg}, V_I \rangle) = \emptyset.$$

Whether data privacy is preserved can be decided by constructing a *canonical* knowledge base that contains precisely the information contained in $\mathcal{O}_{bg}$ and V_I, and querying q on it.

Definition Given a knowledge base $\mathcal{O}_{bg}$ and a view V_I, the canonical knowledge base $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ is defined as

$$\begin{aligned} \mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle} := \mathcal{O}_{bg} \cup & \\ & \{\top \sqsubseteq C \mid \langle \top \sqsubseteq C, \{\mathsf{tt}\} \rangle \in V_I\} \cup \\ & \{a : C \mid \text{there is a set } \mathsf{In} \text{ with } \langle C, \mathsf{In} \rangle \in V_I \text{ and } a \in \mathsf{In}\}. \end{aligned}$$

Theorem 2.2 (see [3, Corollary 1]). *Data privacy is preserved for a query* q *wrt. a view* V_I *and a knowledge base* $\mathcal{O}_{bg}$ *if and only if*

$$\mathsf{ans}(q, \mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}) = \emptyset.$$

According to Definition 2, $\mathsf{ans}(q, \mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle})$ can be computed by a number of entailments which is polynomial to the size of $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$. As it has been already stated, the entailment problem is reducible to the consistency problem which is solvable in ExpTime. Moreover, $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ grows polynomially wrt. $\mathcal{O}_{bg}$ and V_I. Therefore, Theorem 2.2 provides an ExpTime decision procedure to the problem of data privacy on views. The problem is also ExpTime-hard as the problem of concept satisfiability wrt. a consistent TBox[5] is polynomially reducible to the problem of data privacy as follows:

Proposition 2.3. *A concept C is unsatisfiable wrt. a* TBox *$\mathcal{T}$ iff data privacy for $\top \sqsubseteq \neg C$ wrt. $\mathcal{T}$ and the empty view is not preserved.*

Corollary 2.4. *The problem of $\mathcal{ALC}$ data privacy for a query wrt. a view and a knowledge base is ExpTime-complete.*

[5] The proof of ExpTime-completeness [20] is not restricted to a consistent TBox. However, the TBox constructed for the hardness proof is (or can be easily modified to be) consistent.

3 Data privacy on view definitions

We begin with extending the previous definitions to the new problem. A possible view is a view entailed by a possible knowledge base:

Definition A view V_I *is based on* a tuple $\langle \mathcal{O}_{bg}, V \rangle$ if it satisfies the following: (i) V_I is a view of V and (ii) $\mathsf{Poss}_{\langle \mathcal{O}_{bg}, V_I \rangle} \neq \emptyset$.

The problem of data privacy on view definitions can be now formally stated as follows:

Definition Data privacy is preserved for q wrt. a tuple $\langle \mathcal{O}_{bg}, V \rangle$ if for every view V_I based on $\langle \mathcal{O}_{bg}, V \rangle$, data privacy is preserved for q wrt. $\langle \mathcal{O}_{bg}, V_I \rangle$. The data privacy problem on view definitions is to decide whether data privacy is preserved for q wrt. $\langle \mathcal{O}_{bg}, V \rangle$.

The problem of data privacy on a view definition is decidable since it is enough to consider only the views entailed by a finite set of knowledge bases $\mathbb{P}$. Given a tuple $\langle \mathcal{O}_{bg}, V \rangle$ and an individual $n_{ew} \notin \mathsf{Ind}(\mathcal{O}_{bg})$, a knowledge base P is *possible* if

1. $P \supseteq \mathcal{O}_{bg}$ and consistent,
2. if $\top \sqsubseteq C \in P$ then $\top \sqsubseteq C \in \mathcal{O}_{bg} \cup V$, and
3. if $a : C \in P$ then $a : C \in \mathcal{O}_{bg}$ or $(a \in \mathsf{Ind}(\mathcal{O}_{bg}) \cup \{n_{ew}\}$ and $C \in V)$.

Then $\mathbb{P}$ is the set of all possible P wrt. $\langle \mathcal{O}_{bg}, V \rangle$ and n_{ew}.

Theorem 3.1. *Data privacy is preserved for q wrt. a tuple $\langle \mathcal{O}_{bg}, V \rangle$ if and only if, for every view V_I of V that is entailed by some $P \in \mathbb{P}$, data privacy is preserved for q wrt. $\langle \mathcal{O}_{bg}, V_I \rangle$.*

The theorem is proved in Section 5. A naive ExpTime decision procedure for this problem can be constructed directly from the above theorem: first compute $\mathbb{P}$ and all views entailed by its knowledge bases, and then decide data privacy on each of these views. Let P^+ be the knowledge base constructed from $\mathcal{O}_{bg}$ and V as follows:

$$P^+ = \{\top \sqsubseteq C \in V\} \cup \bigcup \{a : C \mid (a \in \mathsf{Ind}(\mathcal{O}_{bg}) \cup \{n_{ew}\}) \text{ and } C \in V\}.$$

Then, $\mathbb{P}$ can be constructed by first computing all subsets of P^+ and then checking their consistency wrt. $\mathcal{O}_{bg}$. Since P^+ can be constructed polynomially wrt. the size of $\mathcal{O}_{bg}$ and V, there are at most $2^{p(n)}$ subsets of P^+ of

maximal cardinality $p(n)$, where n is the total size of $\mathcal{O}_{bg}, V$ and q. Since consistency is decidable in ExpTime, computing $\mathbb{P}$ stays in ExpTime. Now, in order to compute the views entailed by some $P \in \mathbb{P}$, a polynomial number of entailments on every $P \in \mathbb{P}$ is required. Therefore the computation of all views stays also in ExpTime. Finally, Corollary 2.4 together with the fact that V_I grows polynomially wrt. the size of V and P, imply that the total time required for checking privacy on all of the (at most) exponentially many views is again exponential wrt. n.
The problem of data privacy on view definitions is also ExpTime-hard as the corresponding problem on views is polynomially reducible to this problem: data privacy for q is preserved wrt. $\mathcal{O}_{bg}$ and V_I iff it is preserved wrt. $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ and the empty view definition.

Theorem 3.2. *The problem of $\mathcal{ALC}$ data privacy on view definitions is ExpTime-complete.*

In the sequel we present a condition on $\mathcal{O}_{bg}, V$ and q which can be decided in PTime and implies data privacy for q wrt. $\langle \mathcal{O}_{bg}, V \rangle$. Thus, we have a sufficient condition for data privacy that can be checked efficiently. It is based on the syntactic structure of the concepts that constitute the background knowledge and the view definition. We begin by excluding some 'common sense' queries from being potential secrets, because of their trivial (partial) answers.

Definition A *query q is trivial wrt. a tuple $\langle \mathcal{O}_{bg}, V \rangle$* when

- $ans(q, \emptyset) = \{\mathsf{tt}\}$ (i.e. $\emptyset \models q$), if q is a boolean query
- $ans(\top \sqsubseteq q, \emptyset) = \{\mathsf{tt}\}$, if q is a retrieval query and in addition $(\mathsf{Ind}(\mathcal{O}_{bg}) = \emptyset) \Rightarrow (\exists_{C \in V}\ \mathcal{O}_{bg} \not\models \top \sqsubseteq \neg C)$.

A retrieval query might violate privacy only if some individuals are (potentially) given in public. This is the reason for the condition posed on retrieval queries in the above definition. An $\mathcal{ALC}$ query qualifies as a *privacy condition on a tuple* $\langle \mathcal{O}_{bg}, V \rangle$ if it is not trivial wrt. $\langle \mathcal{O}_{bg}, V \rangle$.
Next, we define the boolean function $s_{afe}()$ that decides whether a concept D or a role R exhibits some information about q. Given a knowledge base $\mathcal{O}_{bg}$, a view definition V and a privacy condition q on $\langle \mathcal{O}_{bg}, V \rangle$, the information about a concept D is *safe* if $s_{afe}(D, q)$ returns 1; and the information of a role R is safe if $s_{afe}(R, \langle \mathcal{O}_{bg}, V, q \rangle)$ returns 1.
In the sequel, we use the following conventions. *Concepts and roles of a tuple* $\langle \mathcal{O}_{bg}, V \rangle$ are all inclusion and assertional concepts, assertional roles

and retrieval queries that appear in $\mathcal{O}_{bg}$ or V. If a concept C_2 has a subterm C_1 then C_2 is also written as $C_2[C_1]$. If, in addition, there is an occurrence of C_1 in C_2 that is not prefixed with a quantifier, then C_2 may also be written as $C_2[C_1]^0$. Similarly, if we want to emphasize that C_1 is not prefixed in C_2 with an existential quantifier, then C_2 may also be written as $C_2[C_1]^{0^\exists}$. For example, the concept $A_1 \sqcup \forall R_2.\neg A_2$ can be also written as $A_1 \sqcup \forall R_2.\neg A_2[\neg A_2]$ or as $A_1 \sqcup \forall R_2.\neg A_2[\neg A_2]^{0^\exists}$ but not as $A_1 \sqcup \forall R_2.\neg A_2[\neg A_2]^0$.
Now, assume we are given a query q^{c} where C is the inclusion or assertional concept of q (i.e. $q^{\mathsf{c}} = \top \sqsubseteq C$ or $q^{\mathsf{c}} = C$). The function $s_{afe}()$ is defined on concepts and roles as follows:
For a concept D, $s_{afe}(D, q^{\mathsf{c}}) = 1$ iff there are no D_1 and C_1 subterms of D and C, respectively, of the form:

a. $D_1 = C_1 = A,$ or

b. $D_1 = C_1 = \neg A,$ or

c. $D_1 = QR.D_2$ and $C_1 = QR.C_2,$

where $A \in$ AConc, $R \in$ Rol and $Q \in \{\forall, \exists\}$, and for which either

1. $D[D_1]^0$ and $C[C_1]^{0^\exists}$ hold, or
2. $D[D_1]^0$, $C[\exists R.C'[C_1]]^{0^\exists}$ and $C[\forall R.C'']$ hold.

For a role R and a tuple $\langle \mathcal{O}_{bg}, V\rangle$, $s_{afe}(R, \langle \mathcal{O}_{bg}, V, q^{\mathsf{c}}\rangle) = 1$ iff:

1. C is not of the form $C[\exists R.C']^0$ and
2. for every concept D_2 for which $D_1[\forall R.D_2]^{0^\exists}$ is a concept of $\langle \mathcal{O}_{bg}, V\rangle$, $s_{afe}(D_2, q^{\mathsf{c}}) = 1.$

Theorem 3.3. *Given a consistent $\mathcal{ALC}$ knowledge base $\mathcal{O}_{bg}$, a view definition V and a privacy condition q on $\langle \mathcal{O}_{bg}, V\rangle$, data privacy is preserved for q wrt. $\langle \mathcal{O}_{bg}, V\rangle$ if for every concept D and role R of $\langle \mathcal{O}_{bg}, V\rangle$*

$$s_{afe}(D, q) = s_{afe}(R, \langle \mathcal{O}_{bg}, V, q\rangle) = 1.$$

The proof of the theorem is presented in Section 5. Given a concept D and a query q^{c}, $s_{afe}(D, q^{\mathsf{c}})$ can be computed as follows: find all occurrences of positive atoms A, negated atoms $\neg A$, universal and existential role restrictions $\forall R$ and $\exists R$, respectively, that appear in D and are not prefixed by a quantifier, and check whether any of them appear also in C. If there are such occurrences which are not prefixed by an existential quantifier in C then $s_{afe}(D, q^{\mathsf{c}}) = 0$. Otherwise, let R' be any of the outermost existentially

restricted roles that prefix some of the above occurrences in C. If R' is also a universal restriction in C then, again, $s_{afe}(D, q^{\text{c}}) = 0$. In all other cases $s_{afe}(D, q^{\text{c}}) = 1$. Finding all the above occurrences takes linear time wrt. the size of D since, at worst all subterms of D will be checked. Checking C for a specific occurrence takes again linear time and thus, the total computation stays in PTime wrt. the size of C and D.
Given a role R and a tuple $\langle \mathcal{O}_{bg}, V, q^{\text{c}} \rangle$, $s_{afe}(R, \langle \mathcal{O}_{bg}, V, q^{\text{c}} \rangle)$ can be computed by a number of $s_{afe}()$ computations on concepts, which are as many as there are concepts of the form $D_1[\forall R.D_2]^{0^{\exists}}$ that occur in $\langle \mathcal{O}_{bg}, V \rangle$. Finding these concepts takes linear time wrt. the size of $\langle \mathcal{O}_{bg}, V \rangle$. Thus, the $s_{afe}()$ function on a role can be computed in PTime, too.
To conclude, deciding data privacy for a privacy condition q wrt. $\langle \mathcal{O}_{bg}, V \rangle$ using the above functions takes polynomial time wrt. the size of q and $\langle \mathcal{O}_{bg}, V \rangle$.

Theorem 3.4. *Given a knowledge base $\mathcal{O}_{bg}$, a view definition V and a privacy condition q, it can be decided in PTime whether for every concept D and role R of $\langle \mathcal{O}_{bg}, V \rangle$ we have $s_{afe}(D, q) = s_{afe}(R, \langle \mathcal{O}_{bg}, V, q \rangle) = 1$.*

As mentioned already above, the solution proposed in Theorem 3.3 is partial. It can correctly detect that data privacy is preserved, for instance, for A wrt. $\langle \{R_1(a, b), R_2(b, c)\}, \{\forall R_1 \exists R_2 A\} \rangle$. However, it cannot detect that data privacy is preserved for A wrt. $\langle \{R_1(a, b), R_2(c, d)\}, \{\forall R_1 \forall R_2 A\} \rangle$ or even for $A \sqcap B$ wrt. $\langle \emptyset, \{A\} \rangle$. In the first case this is because we do not consider individuals at all, in the second case because we do not check whether one of the conjuncts forms a trivial query.
The remainder of the paper is concerned with the proofs of Theorems 3.1 and 3.3.

4 The labelled deductive system

The consistency of an $\mathcal{ALC}$ knowledge base can be decided with the help of tableaux systems [20, 21, 22]. The labelled deductive system $S_{\mathcal{ALC}}$ presented below corresponds to the usual labelled tableaux system for $\mathcal{ALC}$ knowledge bases. It derives sequents of the form $\Gamma \ ; \ \hat{T}$ where Γ is a multiset of assertions and $\hat{T}$ is an optional concept. Generally speaking, Γ corresponds to the information of an ABox while $\hat{T}$ to the information of a TBox. If such a sequent is provable in $S_{\mathcal{ALC}}$, then the corresponding knowledge base is inconsistent.

The system $S_{\mathcal{ALC}}$ consists of the following left-hand sided rules where the schematic letters x, y stand for individuals, A for an atomic concept, C and D for arbitrary concepts, and R for a role.

$$\frac{}{x : A,\ x : \neg A,\ \Gamma \ ; \ \hat{T}}\ (ax),$$

$$\frac{x : \underline{\hat{T}},\ \Gamma \ ; \ \underline{\hat{T}}}{\Gamma \ ; \ \underline{\hat{T}}}\ (GCI) \quad \text{where } x \text{ appears in } \Gamma \text{ and } x : \hat{T} \notin \Gamma,$$

$$\frac{x : \underline{C},\ x : \underline{D},\ x : \underline{C \sqcap D},\ \Gamma \ ; \ \hat{T}}{x : \underline{C \sqcap D},\ \Gamma \ ; \ \hat{T}}\ (\sqcap) \quad \text{where } \{x : C,\ x : D\} \nsubseteq \Gamma,$$

$$\frac{x : \underline{C},\ x : \underline{C \sqcup D},\ \Gamma \ ; \ \hat{T} \qquad x : \underline{D},\ x : \underline{C \sqcup D},\ \Gamma \ ; \ \hat{T}}{x : \underline{C \sqcup D},\ \Gamma \ ; \ \hat{T}}\ (\sqcup)$$

where $\{x : C,\ x : D\} \cap \Gamma = \emptyset$,

$$\frac{y : \underline{C},\ (x,y) : \underline{R},\ x : \underline{\exists R.C},\ \Gamma \ ; \ \hat{T}}{x : \underline{\exists R.C},\ \Gamma \ ; \ \hat{T}}\ (\exists)$$

where $\{(x,z) : R,\ z : C\} \nsubseteq \Gamma$ for any z and y is fresh,

$$\frac{y : \underline{C},\ x : \underline{\forall R.C},\ (x,y) : R,\ \Gamma \ ; \ \hat{T}}{x : \underline{\forall R.C},\ (x,y) : R,\ \Gamma \ ; \ \hat{T}}\ (\forall) \quad \text{where } y : C \notin \Gamma.$$

If $a : C$ (or $(a,b) : R$) is an assertion of a sequent S then C (or R) is called *entity* of S and a (or (a,b)) is its *label*. The single concept $\hat{T}$ is also an entity of S. The entities that are explicitly stated in a rule are called *active entities*. The entity $\hat{T}$ is active only in (GCI).

We colour every entity of a sequent by exactly one colour.[6] This is an information that is useful in view of the privacy setting and will be used

[6] For the printed version, instead of colouring, entities are prefixed with a symbol, e.g. $?C$ or $!C$.

later on to distinguish public information from private one. If all entities of a sequent are coloured the same, then the colour is omitted. Also, a coloured Γ denotes that all entities of Γ are coloured the same.
It is convenient to colour also rule applications according to the colours of their active concepts. Rule applications can be then single-coloured or mixed. A rule application is *well-coloured* if every entity that appears in the conclusion has the same colour as its duplication in the premise, and the entity that is underlined in the conclusion (as shown in the rules above) has the same colour as all underlined entities in the premise.
A *coloured derivation* Δ is a tree of well-coloured rule applications. The sequent that appears at the root of Δ is its *conclusion* whereas the sequents on its leaves are its *premises*. Finally, a coloured $S_{\mathcal{ALC}}$ *proof* of a sequent S is a coloured derivation in $S_{\mathcal{ALC}}$ with conclusion S and all of its premises being empty. For example, the following is a bi-coloured proof:

$$\dfrac{\dfrac{}{a_1 : !\forall R.C,\ h_3 : !C,\ h_3 : ?\neg C,\ (a_1, h_3) : !R\ \ ;}\ (ax)}{a_1 : !\forall R.C,\ h_3 : ?\neg C,\ (a_1, h_3) : !R\ \ ;}\ (\forall)$$

Definition Let $\mathcal{O}$ be a knowledge base with a non-empty ABox $\mathcal{A}$ and a TBox $\mathcal{T} = \{\top \sqsubseteq C_i \mid 0 \leq i \leq n\}$. Then, we say that $\mathcal{O}$ *has a proof in* $S_{\mathcal{ALC}}$ if there is an $S_{\mathcal{ALC}}$ proof of the sequent $\mathcal{A}\ \ ;\ \ \hat{T}$ where

$$\hat{T} = \prod_{0 \leq i \leq n} C_i \quad .$$

The following theorem restates the well-known decision procedure result for the consistency of an $\mathcal{ALC}$ ABox with respect to an $\mathcal{ALC}$ TBox.

Theorem 4.1 (see for instance [20])**.** *An $\mathcal{ALC}$ knowledge base with a non-empty* ABox *is inconsistent iff it has a proof in $S_{\mathcal{ALC}}$.*

5 Proving privacy

Lemma 5.1. *Let S be a sequent of the form $\Gamma_x,\ \Gamma\ \ ;\ \ \hat{T}$, where Γ_x is the least multiset of assertions satisfying the following conditions:*

- *if $x : C$, $(x, x') : R$ or $(x', x) : R$ is an assertion in S, then this is in Γ_x,*
- *for every x' that appears in Γ_x, if $x' : C$, $(x', x'') : R$ or $(x'', x') : R$ is an assertion in S then this is in Γ_x.*

If S is provable in $S_{\mathcal{ALC}}$ then the sequent $\Gamma_x\ \ ;\ \ \hat{T}$ or $\Gamma\ \ ;\ \ \hat{T}$ is also provable.

Proof Let Π be a proof of S. We prove the theorem by induction on the length l of Π.
Base case: $l = 1$. Then $S = y : A,\ y : \neg A,\ \Delta\ \ ;\ \ \hat{T}$. If $y : A \in \Gamma_x$ then also $y : \neg A \in \Gamma_x$ and so, $\Gamma_x\ \ ;\ \ \hat{T}$ is provable. Otherwise, if $y : A \in \Gamma$, then also $y : \neg A \in \Gamma$. But then, $\Gamma\ \ ;\ \ \hat{T}$ is provable.
Induction step. We assume that the theorem holds for proofs of length n. By a case analysis on the rule application r of Π that concludes S, we show that the theorem also holds for proofs of length $n + 1$:

- $r = GCI$. Then the premise of r is $S' = y : \hat{T},\ \Gamma_x,\ \Gamma\ \ ;\ \ \hat{T}$, where y appears in S and $y : \hat{T} \notin S$. By the definition of S, y appears in exactly one of Γ_x and Γ. Adding the new assertion to the multiset y appears in, gives a sequent that matches the preconditions of the theorem, and S' takes precisely that form. Therefore, the induction hypothesis applies to S'. Again, we distinguish between the possible locations of y:
  - If y appears in Γ_x, then the induction hypothesis on S' results a proof of $y : \hat{T},\ \Gamma_x\ \ ;\ \ \hat{T}$ or a proof of $\Gamma\ \ ;\ \ \hat{T}$. Applying GCI to the first sequent results a proof of $\Gamma_x\ \ ;\ \ \hat{T}$. Therefore, in both cases the theorem has been shown.
  - If y appears in Γ, then the induction hypothesis on S' results a proof of $\Gamma_x\ \ ;\ \ \hat{T}$ or a proof of $y : \hat{T},\ \Gamma\ \ ;\ \ \hat{T}$. Applying GCI to the latter results a proof of $\Gamma_x\ \ ;\ \ \hat{T}$ or a proof of $\Gamma\ \ ;\ \ \hat{T}$, as required.
- $r = \sqcap$. Then $S = y : C_1 \sqcap C_2,\ \Delta\ \ ;\ \ \hat{T}$ and $\{y : C_1, y : C_2\} \nsubseteq \Delta$. The premise of r is $S' = y : C_1,\ y : C_2,\ y : C_1 \sqcap C_2,\ \Delta\ \ ;\ \ \hat{T}$. As in the previous case, the induction hypothesis applies to S' when the new assertions are added to the multiset that contains $y : C_1 \sqcap C_2$. We distinguish between the possible locations of these assertions:

- If $y : C_1 \sqcap C_2 \in \Gamma_x$, then the induction hypothesis on S' yields a proof of $y : C_1,\ y : C_2,\ \Gamma_x\ \ ;\ \ \hat{T}$ or a proof of $\Gamma\ \ ;\ \ \hat{T}$. Applying the ($\sqcap$)-rule to the first sequent yields a proof of $\Gamma_x\ \ ;\ \ \hat{T}$ and completes the required results.
- If $y : C_1 \sqcap C_2 \in \Gamma$, then the induction hypothesis on S' yields a proof of $y : C_1,\ y : C_2,\ \Gamma\ \ ;\ \ \hat{T}$ or a proof of $\Gamma_x\ \ ;\ \ \hat{T}$. Applying the ($\sqcap$)-rule to the first sequent yields a proof of $\Gamma\ \ ;\ \ \hat{T}$, as required.

- $r = \sqcup$. Then $S = y : C_1 \sqcup C_2,\ \Delta\ \ ;\ \ \hat{T}$ and $\{y : C_1, y : C_2\} \cap \Delta = \emptyset$. The premises of r are

$$S_1 = y : C_1,\ y : C_1 \sqcup C_2,\ \Delta\ \ ;\ \ \hat{T} \qquad \text{and}$$
$$S_2 = y : C_2,\ y : C_1 \sqcup C_2,\ \Delta\ \ ;\ \ \hat{T}.$$

As in the previous case, the induction hypothesis applies to both S_1 and S_2 when, in each of the cases, the new assertion is added to the multiset that contains $y : C_1 \sqcup C_2$. We distinguish between the possible locations of these assertions:

  - If $y : C_1 \sqcup C_2 \in \Gamma_x$, then the induction hypothesis on S_1 yields a proof of $y : C_1,\ \Gamma_x\ \ ;\ \ \hat{T}$ or a proof of $\Gamma\ \ ;\ \ \hat{T}$. And the induction hypothesis on S_2 yields a proof of $y : C_2,\ \Gamma_x\ \ ;\ \ \hat{T}$ or a proof of $\Gamma\ \ ;\ \ \hat{T}$. Thus, there is either a proof of $\Gamma\ \ ;\ \ \hat{T}$ or there are the proofs of $y : C_1,\ \Gamma_x\ \ ;\ \ \hat{T}$ and $y : C_2,\ \Gamma_x\ \ ;\ \ \hat{T}$. Applying the ($\sqcup$)-rule to these sequents yields a proof of $\Gamma_x\ \ ;\ \ \hat{T}$ which completes the required results.
  - If $y : C_1 \sqcup C_2 \in \Gamma$, then the proof is similar to the previous case.

- $r = \exists$. Then $S = y : \exists R.C,\ \Delta\ \ ;\ \ \hat{T}$ and the premise of r is

$$S' = z : C,\ (y, z) : R,\ y : \exists R.C,\ \Delta\ \ ;\ \ \hat{T},$$

where z is fresh. Since z does not appear in Δ, adding the new assertions to the multiset that contains $y : \exists R.C$ yields a sequent that satisfies the preconditions of the theorem, and therefore the induction hypothesis applies to S'. The case distinction is similar to that of the previous rules.

- $r = \forall$. Then $S = y : \forall R.C,\ (y, z) : R,\ \Delta\ \ ;\ \ \hat{T}$ and the premise of r is $S' = z : C,\ y : \forall R.C,\ (y, z) : R,\ \Delta\ \ ;\ \ \hat{T}$. By the definition of S, $(y, z) : R$ is in the same multiset $y : \forall R.C$ is in, and z does not appear in the other multiset. This implies that S' satisfies the preconditions and so the induction hypothesis applies to it. The case distinction is similar to that of the previous rules.

Lemma 5.2. *Let $\mathbb{P}$ be the set of possible knowledge bases wrt. a tuple $\langle \mathcal{O}_{bg}, V \rangle$. If $P \in \mathbb{P}$ and V_I is the view of V entailed by P, then $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ and P are logically equivalent.*

Proof First, we show that every element of $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ is entailed by P and therefore P is at least as strong as $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$. Since $P \supseteq \mathcal{O}_{bg}$, the elements of $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ that come from $\mathcal{O}_{bg}$ are entailed by P. The rest of the elements come from V_I which, by definition, is a view entailed by P and so each of these elements is also entailed by P.

Second, we show that $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle} \supseteq P$ and therefore $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ is at least as strong as P. Since $\mathcal{O}_{bg} \subseteq \mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$, the elements of P that come from $\mathcal{O}_{bg}$ are also in $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$. The rest of the elements come from V. Now, since V_I is a view of V entailed by P, we have that for every inclusion axiom $\top \sqsubseteq C \in P \setminus \mathcal{O}_{bg}$ there is a tuple $\langle \top \sqsubseteq C, \{\mathsf{tt}\} \rangle \in V_I$. Similarly, for every assertion $a : C \in P \setminus \mathcal{O}_{bg}$ there is a tuple $\langle C, \mathsf{In} \rangle \in V_I$ with $a \in \mathsf{In}$. Therefore, these elements are also in $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$.

Theorem 3.1. *Data privacy is preserved for q wrt. a tuple $\langle \mathcal{O}_{bg}, V \rangle$ if and only if, for every view V_I of V that is entailed by some $P \in \mathbb{P}$, data privacy is preserved for q wrt. $\langle \mathcal{O}_{bg}, V_I \rangle$.*

Proof ($\Rightarrow$) Trivial.

($\Leftarrow$) We prove the contrapositive. Assume that V_I is a view based on $\langle \mathcal{O}_{bg}, V \rangle$ on which q is not preserved. As a consequence of Theorem 2.2, $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle} \models q$, if q is boolean or $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle} \models d : q$, for some $d \in \mathsf{Ind}(\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle})$, if q is retrieval.

Let $\mathcal{T}$ be the TBox of $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ and $\mathcal{A}$ its ABox. First we show that q (resp. $d : q$) is entailed by a subset of $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ that contains at most one additional individual (i.e. an individual that does not appear in $\mathcal{O}_{bg}$). Assume that there are more than one such individuals appearing in V_I. We distinguish between the possible forms of q:

- $q = \top \sqsubseteq C$. We show that $\mathcal{T} \models \top \sqsubseteq C$ also holds. We have that $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle} \models \top \sqsubseteq C$, thus $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle} \cup \{a : \neg C\}$ is inconsistent for $a \notin \mathsf{Ind}(\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle})$. Therefore, there is a proof of $a : \neg C,\ \mathcal{A}\ ;\ \hat{T}$ where, $\hat{T}$ is the concept that represents all inclusion axioms of $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$. Since a does not appear in $\mathcal{A}$, by Lemma 5.1 we get a proof of $a : \neg C\ ;\ \hat{T}$ or a proof of $\mathcal{A}\ ;\ \hat{T}$. While the latter is not possible because it implies that $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ is inconsistent, the first proof implies that $\mathcal{T} \cup \{a : \neg C\}$ is inconsistent and so, by Theorem 2.1, $\mathcal{T} \models \top \sqsubseteq C$.

- $q = C$. Adding $d : \neg C$ to $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ would cause inconsistency and so, there is a proof Π of $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle} \cup \{d : \neg C\}$. Let Γ_x be the set of assertions of one of the additional individuals $x \neq d$. Note that x does not appear in any role assertion in $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$. Therefore, Lemma 5.1 applies to Π with such a Γ_x. This gives either a proof of $\Gamma_x \;\; ; \;\; \hat{T}$, where $\hat{T}$ is the concept that represents all inclusion axioms of $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$, or a proof of $\Gamma, \; d : \neg C \;\; ; \;\; \hat{T}$, where $\Gamma = \mathcal{A} \backslash \Gamma_x$. While the first proof is not possible since it would imply that $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ is inconsistent, the second proof implies that there is a subset of $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ with one additional individual less, that also entails $d : q$. Applying the lemma iteratively to the above proof results a knowledge base that contains at most one additional individual.

Let $C' \subseteq \mathcal{C}_{\langle \mathcal{O}_{bg}, V_I \rangle}$ be the obtained knowledge base that has at most one additional individual x. Renaming every occurrence of x in C' by n_{ew} results in a knowledge base, say C^r, which is equivalent to C' modulo individual renaming. Therefore C^r also entails some private data, and so does $C^r \cup \mathcal{O}_{bg}$, too. The latter is a knowledge base in $\mathbb{P}$. Let V_I^r be the view of V that is entailed by $C^r \cup \mathcal{O}_{bg}$. Then, Lemma 5.2 results that $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I^r \rangle}$ is equivalent to $C^r \cup \mathcal{O}_{bg}$ and so, data privacy for q is not preserved on V_I^r either.

We now present some results on coloured $S_{\mathcal{ALC}}$ proofs which will be used in the proof of Theorem 3.3.

Lemma 5.3. *Assume that we are given a query* $q^{\mathcal{C}}$ *and a bi-coloured* $S_{\mathcal{ALC}}$ *proof* Π *of a sequent* $S_1 = d : ?\neg C, \; !\Gamma \;\; ; \;\; !\hat{T}$*. Furthermore, assume that*

(i) $s_{afe}(!R, \langle \Gamma \cup \{\top \sqsubseteq \hat{T}\}, \emptyset, q^{\mathcal{C}} \rangle) = s_{afe}(!D, q^{\mathcal{C}}) = 1$*, for all entities* $!R$ *and* $!D$*, respectively, in* S_1*.*

Let S_2 *be a sequent in* Π *of the form*

(ii) $x : ?rd, \; x : !gr, \; \Delta \;\; ; \;\; !\hat{T}$ *with*

(iii) $?rd = C_1[A]^0$ *and* $!gr = D_1[\neg A]^0$*, or*
$?rd = C_1[\neg A]^0$ *and* $!gr = D_1[A]^0$*, or*
$?rd = C_1[QR.C_2]^0$ *and* $!gr = D_1[\neg QR.D_2]^0$ *for* $Q \in \{\forall, \exists\}$*.*

Then, there is a mixed-rule application in the path between S_1 *and* S_2*.*

Proof By induction on the length n of the path between S_1 and S_2. Base case: $n = 0$. Then $S_1 = S_2$ and so $C = \neg rd$. By the definition of $s_{afe}()$ on concepts (first condition), $s_{afe}(gr, q^{\neg rd}) = 0$ for every case of (iii) and so, (i) is contradicted. Therefore, this is not possible.

Induction step: assume that there are $n+1$ rule applications between S_1 and S_2 and that all of them are single-coloured. Let r be the rule application with premise S_2 and conclusion S_2'. By a case analysis on r we show that in all possible cases, S_2' satisfies (ii) and (iii). Thus by the induction hypothesis there is a mixed-rule application between S_1 and S_2.
If both $?rd$ and $!gr$ are in S_2' then S_2' satisfies (ii) and (iii). Otherwise, one of the two is an active entity in S_2 that does not appear in S_2'. There are the following cases on r:

- $r = (?GCI)$. This case is not possible.
- $r = (!GCI)$. Then $S_2' = x : ?rd,\ \Delta \ ; \ !gr$. Since $?rd$ appears in S_2', by the form of S_1 we have that $\neg rd$ is a subterm of C. There are two cases on C:
  - $C = C[\neg rd]^{0^{\exists}}$. This is not possible as - in all cases of (iii) - it would imply that $s_{afe}(!gr, q^{\mathsf{c}}) = 0$, which contradicts (i) ($!gr$ appears in S_1). To see this, consider for instance the case when $?rd = C_1[A]^0$ and $!gr = D_1[\neg A]^0$. Then C is of the form $C[\neg A]^{0^{\exists}}$. Therefore, in the definition of $s_{afe}()$ on concepts, there are subterms of $!gr$ and C such that b. and 1. hold and so $s_{afe}(!gr, q^{\mathsf{c}}) = 0$. The other cases are similar.
  - C is of the form $C[\exists R.C'[\neg rd]]^{0^{\exists}}$ and not of the form $C[\neg rd]^{0^{\exists}}$. This implies that $z : ?\forall R.\neg C'$ is an active entity on a rule below r and so, since all rules below r are single-coloured, there is an $?R$ entity in Π. By the form of S_1, this is possible only if there is an entity $?\exists R.C''$ in Π, which means that C is also of the form $C[\forall R.\neg C'']$. This however - in all cases of (iii) - contradicts $s_{afe}(!gr, q^{\mathsf{c}}) = 1$. When, for instance, $?rd = C_1[A]^0$ and $!gr = D_1[\neg A]^0$, C takes the form $C[\exists R.C'[\neg A]]^{0^{\exists}}$. Thus, in the definition of $s_{afe}()$ on concepts, there are subterms of $!gr$ and C such that b. and 2. hold and so $s_{afe}(!gr, q^{\mathsf{c}}) = 0$. The other cases are similar.
- $r \in \{(\sqcap), (\sqcup)\}$. Then $S_2' = x : ?C'[rd]^0,\ x : !D'[gr]^0,\ \Delta' \ ; \ !\hat{T}$. Both C' and D' qualify as $?rd$ and $!gr$, respectively, and so S_2' satisfies (ii) and (iii).
- $r = (\exists)$. This cannot be the case, since the active concept that does not appear in the conclusion has to have a fresh label. Therefore, not both $!gr$ and $?rd$ can have the same label.

- $r = (?\forall)$. Then $S'_2 = y : ?\forall R.rd,\ (y,x) : ?R,\ x : !gr,\ \Delta'\ \ ;\ \ !\hat{T}$. Since $(y,x) : ?R$ cannot occur in S_1, this assertion was created by an $(?\exists)$-rule below r, and therefore $y : ?\exists R.C'$ is an active entity in a rule below S'_2 and x is fresh. Since all rules below r are single-coloured and x is fresh, $x : !gr$ can appear in S_2 only in the case $!\hat{T}$ is of the form $!\hat{T}[gr]^0$. Reasoning is then continued similarly to the $!GCI$ case.

- $r = (!\forall)$. Then, $S'_2 = y : !\forall R.gr,\ (y,x) : !R,\ x : ?rd,\ \Delta'\ \ ;\ \ !\hat{T}$. If $(y,x) : !R$ were created by an $(!\exists)$-rule, then x would be a new label and, because of the single-coloured rules below S'_2, $x : ?rd$ would not be possible. Therefore,

  $$(y,x) : !R \text{ appears in } S_1. \tag{1}$$

  Furthermore, the presence of $!\forall R.gr$ implies that there is an entity of the form $!D'[\forall R.gr]$ in S_1.

  - If there is such an entity of the form $!D'[\forall R.gr]^{0^\exists}$, by (1) and (i) we have $s_{afe}(!R, \langle \Gamma \cup \{\top \sqsubseteq \hat{T}\}, \emptyset, q^{\mathsf{c}}\rangle) = 1$ and $!D'[\forall R.gr]^{0^\exists}$ is an entity in Γ. So, by the definition of $s_{afe}()$ on roles $s_{afe}(gr, q^{\mathsf{c}}) = 1$. Therefore, by the definition of $s_{afe}()$ on concepts and (iii), we have that C cannot be of the form $C[\neg rd]^{0^\exists}$ (for details see the first case of $!GCI$). However, by S_1 we have that $C[\neg rd]$ and thus, there is an active entity $?\forall R'.C'[rd]^0$ below r. Therefore, if w is the label of this entity, we have that $(w,x) : ?R'$ appears in a sequent below S'_2 (since $x : ?rd$ is in S'_2). Again, this assertion must have been created by an $(?\exists)$-rule and thus x is fresh which contradicts (1).
  - Otherwise, we find that for every $z_i : !D''[\forall R.gr]^{0^\exists}$ in Π, z_i is a fresh variable. Since y is a label of such a D'' (note that $y : !\forall R.gr$ occurs in S'_2), y is also fresh and so (1) is contradicted.

By induction on the length of the derivation Π, we can show the following:

Lemma 5.4. *Let Π be a bi-coloured proof of the sequent*

$$!\Gamma, ?\Delta\ \ ;\ \ !\hat{T}$$

that has only single-coloured rule applications. Then, there is either a proof of the sequent $!\Gamma\ \ ;\ \ !\hat{T}$ or a proof of the sequent $?\Delta\ \ ;\ \ $.

We are now ready to prove Theorem 3.3.

Theorem 3.3. *Given a consistent $\mathcal{ALC}$ knowledge base $\mathcal{O}_{bg}$, a view definition V and a privacy condition q on $\langle \mathcal{O}_{bg}, V\rangle$, data privacy is preserved for q wrt. $\langle \mathcal{O}_{bg}, V\rangle$ if for every concept D and role R of $\langle \mathcal{O}_{bg}, V\rangle$*

$$s_{afe}(D, q) = s_{afe}(R, \langle \mathcal{O}_{bg}, V, q\rangle) = 1.$$

Proof By contradiction. Let $q = \top \sqsubseteq C$ or $q = C$. Assume that (a) there is a V_I on $\langle \mathcal{O}_{bg}, V\rangle$ such that data privacy is not preserved for q with respect to V_I while (b) $s_{afe}(D, q) = s_{afe}(R, \langle \mathcal{O}_{bg}, V, q\rangle) = 1$, for all concepts and roles D and R, respectively, of $\langle \mathcal{O}_{bg}, V\rangle$.
Applying Theorem 2.2 to assumption (a) yields $ans(q, \mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle}) \neq \emptyset$. That is, $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle} \models \top \sqsubseteq C$ if q is boolean and $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle} \models d : C$ for some $d \in \mathsf{Ind}(\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle})$, if q is retrieval. Thus, the knowledge base $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle} \cup \overline{q}$ is inconsistent, where $\overline{q}$ is given as follows:

$$\begin{aligned} \overline{\top \sqsubseteq C} &:= \{d' : \neg C\}, \text{ for some fixed } d' \in \mathsf{Ind}, \\ \overline{C} &:= \{d : \neg C\} \ . \end{aligned}$$

Theorem 4.1 implies that the knowledge base $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle} \cup \overline{q}$ has a proof in $S_{\mathcal{ALC}}$ and thus, the sequent $\Gamma,\ \overline{q}\ \ ;\ \ \hat{T}$ is provable in $S_{\mathcal{ALC}}$, where Γ and $\hat{T}$ are the ABox and the TBox transformation of the canonical knowledge base $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle}$. We distinguish between public and private information in the sequent by colouring the entities derived from $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle}$ green (resp. !) and the entity of $\overline{q}$ red (resp. ?). Let Π be a bi-coloured proof of

$$!\Gamma,\ ?\overline{q}\ \ ;\ \ !\hat{T}\ . \tag{2}$$

According to the colours of its rule applications (green, red or mixed), Π has either at least one mixed rule or it has no mixed rule at all. We distinguish between these two cases:

1. Π has no mixed rule application. Then, Lemma 5.4 applies and yields either a proof of $!\Gamma\ \ ;\ \ !\hat{T}$ or a proof of $?\overline{q}\ \ ;\ $. This means that either $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle}$ or $\overline{q}$ is an inconsistent knowledge base. Since $\mathcal{O}_{bg}$ is consistent, $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle}$ is also consistent and therefore the first case is not possible. In the second case, the query q can be either boolean or retrieval. In both cases Theorem 2.1 applied on $\overline{q}$ results $\emptyset \models \top \sqsubseteq C$. Furthermore, in the case $q = C$, d is an individual either in $\mathcal{O}_{bg}$ or in V_I, which means that either $\mathsf{Ind}(\mathcal{O}_{bg}) \neq \emptyset$ or that there exists a retrieval query D' in V such that $\mathcal{O}_{bg} \not\models \top \sqsubseteq \neg D'$. Therefore, q is a trivial query and the assumption of the theorem is contradicted.

2. Π has at least one mixed rule. Let r be a mixed-rule for which all rules below r are single-coloured. If we can show that Lemma 5.3 applies with

$S_1 = (2)$ and S_2 the conclusion of r, then there is a mixed rule application below r which contradicts the definition of r and thus the theorem is shown.

Therefore, it remains to prove that the assumptions of Lemma 5.3 hold. First we show that for all entities $!R$ and $!D$ of the sequent (2)

$$s_{afe}(!R, \langle \Gamma \cup \{\top \sqsubseteq \hat{T}\}, \emptyset, q\rangle) = s_{afe}(!D, q) = 1. \tag{3}$$

For $D \in \Gamma$, we also have $D \in \langle \mathcal{O}_{bg}, V\rangle$. Thus $s_{afe}(D, q) = 1$ by the assumptions of the theorem.
For $D = \hat{T}$, let D_1 be a subterm of $\hat{T}$ as required by the cases a. - c. in the definition of $s_{afe}()$. We observe that if $\hat{T}[D_1]^0$ holds, then there is some $D_2 \in \langle \mathcal{O}_{bg}, V\rangle$ of the form $D_2[D_1]^0$ and $s_{afe}(D_2, q) = 1$. This implies $s_{afe}(D, q) = 1$.
By the assumption of the theorem we have $s_{afe}(R, \langle \mathcal{O}_{bg}, V, q\rangle) = 1$. According to the definition of $s_{afe}()$ on roles, the degree of R might change only if there is a concept $D_1[\forall R.D_2]^{0^\exists}$ in $\langle \Gamma \cup \{\top \sqsubseteq \hat{T}\}, \emptyset\rangle$ and there is no concept $D'_1[\forall R.D_2]^{0^\exists}$ in $\langle \mathcal{O}_{bg}, V\rangle$. However, this is not possible since, on one hand, by the construction of $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle}$ all concepts of $\langle \mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle}, \emptyset\rangle$ are also concepts of $\langle \mathcal{O}_{bg}, V\rangle$. Thus, Γ does not introduce any new concepts. On the other hand, $\hat{T}$ is a conjunction of concepts of $\mathcal{C}_{\langle \mathcal{O}_{bg}, V_I\rangle}$ and so, for every concept $D_1[\forall R.D_2]^{0^\exists}$ of $\langle \Gamma \cup \{\top \sqsubseteq \hat{T}\}, \emptyset\rangle$ there is a concept $D'_1[\forall R.D_2]^{0^\exists}$ in $\langle \mathcal{O}_{bg}, V\rangle$. Therefore, we conclude that $s_{afe}(R, \langle \Gamma \cup \{\top \sqsubseteq \hat{T}\}, \emptyset, q\rangle) = 1$ and thus (3) holds.
Next, we show that the conclusion of r has the form required for the sequent S_2 in Lemma 5.3. The mixed rule application r must be of the form:

$$\frac{}{x : !A,\ x : ?\neg A,\ \Gamma' \ ;\ !\hat{T}}\ (ax), \qquad \frac{}{x : ?A,\ x : !\neg A,\ \Gamma' \ ;\ !\hat{T}}\ (ax),$$

$$\frac{y : ?C',\ x : ?\forall R'.C',\ (x,y) : !R',\ \Gamma' \ ;\ !\hat{T}}{x : ?\forall R'.C',\ (x,y) : !R',\ \Gamma' \ ;\ !\hat{T}}\ (\forall),$$

or

$$\frac{y : !C',\ x : !\forall R'.C',\ (x,y) : ?R',\ \Gamma'\ (!\hat{T})}{x : !\forall R'.C',\ (x,y) : ?R',\ \Gamma' \ ;\ !\hat{T})}\ (\forall)$$

where $y : C'$ does not appear in Γ'. If $r = (ax)$ then its conclusion is trivially of the required form. Otherwise, if $r = (\forall)$ we observe that in both applications of $(\forall)$, the role assertion $(x, y) : R'$ was created by an $(\exists)$-rule. This can be seen as follows:

- For the first rule. Since $?\forall R'.C'$ occurs in the conclusion of r, we have either

(i) $\neg C$ is of the form $(\neg C)[\forall R'.C']^0$ or

(ii) $QR''.C''[\forall R'.C']^0$ for some quantifier Q, is an active entity in a rule $?r'$ that appears below r.

We show that in both cases, $(x, y) : !R'$ cannot appear in (2). If (i) holds we have $C[\exists R'.\neg C']^0$.Thus by the definition of $s_{afe}()$ on roles, we have $s_{afe}(R', \langle \mathcal{O}_{bg}, V, q\rangle) = 0$ and so, $!R$ does not appear in (2).

If (ii) holds, then there is an entity $(z, x) : ?R''$ in the premise of r' which, by definition of (2), cannot appear in (2). This implies that $(z, x) : ?R''$ was created by an $(\exists)$-rule and x is fresh. Therefore, $(x, y) : R'$ does not appear in (2).

- For the second rule. By the definition of q, the set $\overline{q}$ does not contain any role assertions.

Consequently, in both cases the role assertion was created in the course of the proof and this can happen only by means of an $(\exists)$-rule application. Thus, $x : !\exists R'.D'$ or $x : ?\exists R'.D'$ appears in the proof before the first or the second $(\forall)$-rule, respectively. Since nothing is thrown away while applying rules, the existential concepts occur in Γ' in their respective rule application above. Therefore, the conclusion of r has the required form and Lemma 5.3 applies to Π with $S_1 =$ (2) and S_2 the conclusion of r.

6 Conclusions

We have studied the problem of data privacy on view definitions for $\mathcal{ALC}$ knowledge bases. Our setting does not assume a completely given knowledge base. We only assume that we are given a view definition and some additional (ontological) knowledge. The goal is to verify that a given privacy condition is preserved for all possible views of the given definition. We have presented two solutions to the problem: the first one is ExpTime-complete whereas the second is only partial and can be decided in PTime. The first solution is obtained by restricting the views that need to be checked against privacy to be finitely many. The second is done by syntactically comparing concepts and roles occurring in the public part with the privacy condition.

From this second solution, we get for instance the following application. Let a safe role be a role for which there are no role assertions given in public. We can establish the privacy of concepts that are built from safe roles and atomic concepts that appear in the public part only behind an existentially quantified role or behind a safe universally quantified role. This pattern applies independently of the internal structure of an ontology and therefore it also applies to modular or $\mathcal{E}$-connected ontologies.

We plan to extend our method to more expressive languages. In particular, we would like to study how nominals, inverse and functional roles behave under privacy. Such a study would allow us to identify a wider range of privacy preserving ontologies.

References

[1] M. K. Smith, C. Welty, and D. L. McGuinness, “OWL web ontology language guide,” 2004. Available at `http://www.w3.org/TR/owl-guide/`.

[2] K. Stoffel and T. Studer, “Provable data privacy,” in *Database and Expert Systems Applications DEXA 2005* (K. Viborg, J. Debenham, and R. Wagner, eds.), vol. 3588 of *LNCS*, pp. 324–332, Springer, 2005.

[3] P. Stouppa and T. Studer, “A formal model of data privacy,” in *Perspectives of System Informatics PSI’06* (I. Virbitskaite and A. Voronkov, eds.), vol. 4378 of *LNCS*, pp. 401–411, Springer, 2007.

[4] R. van der Meyden, “Logical approaches to incomplete information: a survey,” in *Logics for databases and information systems*, pp. 307–356, Kluwer Academic Publishers, 1998.

[5] A. Calì, D. Calvanese, G. D. Giacomo, and M. Lenzerini, “Data integration under integrity constraints,” in *Proc. of CAiSE 2002*, vol. 2348 of *LNCS*, pp. 262–279, Springer, 2002.

[6] A. Y. Halevy, “Answering queries using views: A survey,” *The VLDB Journal*, vol. 10, no. 4, pp. 270–294, 2001.

[7] M. Lenzerini, “Data integration: a theoretical perspective,” in *ACM PODS ’02*, pp. 233–246, ACM Press, 2002.

[8] M. Arenas and L. Libkin, “XML data exchange: Consistency and query answering,” in *PODS*, pp. 13–24, 2005.

[9] R. Fagin, P. G. Kolaitis, R. Miller, and L. Popa, “Data exchange: Semantics and query answering,” *Theoretical Computer Science*, vol. 336, pp. 89–124, 2005.

[10] B. Cuenca Grau, B. Parsia, E. Sirin, and A. Kalyanpur, “Automated partitioning of OWL ontologies using E-connections,” in *Proceedings of Int. Workshop on Description Logics*, 2005.

[11] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev, “E-connections of abstract description systems,” *Artifical Intelligence*, vol. 156, no. 1, pp. 1–73, 2004.

[12] G. Miklau and D. Suciu, “A formal analysis of information disclosure in data exchange,” in *SIGMOD*, 2004.

[13] A. Deutsch and Y. Papakonstantinou, “Privacy in database publishing,” in *ICDT*, 2005.

[14] A. Machanavajjhala and J. Gehrke, “On the efficiency of checking perfect privacy,” in *PODS ’06: Proceedings of Principles of database systems*, pp. 163–172, ACM Press, 2006.

[15] J. Biskup and P. A. Bonatti, “Controlled query evaluation for enforcing confidentiality in complete information systems,” *International Journal of Information Security*, vol. 3, no. 1, pp. 14–27, 2004.

[16] P. Samarati and L. Sweeney, “Generalizing data to provide anonymity when disclosing information (abstract),” in *PODS*, p. 188, ACM Press, 1998.

[17] P. A. Bonatti, S. Kraus, and V. S. Subrahmanian, “Foundations of secure deductive databases,” *Transactions on Knowledge and Data Engineering*, vol. 7, no. 3, pp. 406–422, 1995.

[18] M. Winslett, K. Smith, and X. Qian, “Formal query languages for secure relational databases,” *ACM Trans. Database Syst.*, vol. 19, no. 4, pp. 626–662, 1994.

[19] S. Tobies, *Complexity results and practical algorithms for logics in Knowledge Representation*. PhD thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.

[20] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, eds., *The Description Logic Handbook*. Cambridge University Press, 2003.

[21] F. Baader and U. Sattler, “An overview of tableau algorithms for description logics,” *Studia Logica*, vol. 69, pp. 5–40, 2001.

[22] F. M. Donini and F. Massacci, “EXPTIME tableaux for ALC,” *Artificial Intelligence*, vol. 124, no. 1, pp. 87–138, 2000.