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

Phiniki Stouppa and Thomas Studer

Universität Bern, Institut für Informatik und angewandte Mathematik,
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{stouppa,tstuder}@iam.unibe.ch

**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 the problem of provable privacy 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 condition for data privacy that can be checked in PTime.

## 1 Introduction

In the context of 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.

Such an area is, for example, the semantic web. There, knowledge is represented by ontologies which provide formalizations of concept definitions for an application domain. These ontologies are expressed in an ontology language. OWL (Web Ontology Language) is the W3C endorsed standard language for this purpose. The underlying formal framework of OWL are the so-called description logics [1]. In the present paper we will study the privacy problem with respect to the basic description logic $\mathcal{ALC}$ which is the simplest description logic that is boolean closed.

It was always clear that privacy issues have to be considered in the context of ontology languages. Let us cite the OWL Language Guide [2]: '...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.'

The present paper is the continuation of our work started in [3,4]. There, we introduced the problem of *provable data privacy on views* as follows. Assume that some agent has access to a view provided by an information system. Additionally, there is some background knowledge that is publicly available. The privacy problem under this setting is to decide whether the user is not able to 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*

S. Artemov and A. Nerode (Eds.): LFCS 2009, LNCS 5407, pp. 409–421, 2009.

*answers to $q$ is empty.* If the problem is answered positively, we say that privacy is *preserved* for $q$.

We will now use the notion of provable privacy to study a more general problem: namely, 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 general concept inclusion axioms (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.

Let us now illustrate the difference between privacy on views and privacy on view definitions. Our running example will be a business information system storing information about account managers and their salaries.

*Example 1.* The background knowledge states that an account manager gets a high or a low salary:

$$\texttt{account\_manager} = \texttt{high} \sqcup \texttt{low}.$$

Assume that an agent has access to the views defined by

$$\{\texttt{account\_manager}, \neg\texttt{high}\}$$

and that for some reason the extension of `low` should be hidden.

For the *privacy problem on views*, we assume that we are given the answers to the views. For instance, assume $\{a\}$ is the answer of the query `account_manager` and $\{b\}$ is the answer to the query $\neg$`high`. In this case, privacy for `low` is preserved with respect to the given view, since for no individual we can infer that it belongs to `low`.

For the *privacy problem on view definitions*, we do *not* assume that the answers to the views are given. Rather the question is whether privacy is preserved for all possible sets of answers. In our example, privacy is not preserved on the view definition. Consider the following possibility: the answer to the query `account_manager` might be $\{a, b\}$ and the answer to the query $\neg$`high` might be $\{b\}$. In this case $b$ must belong to `low`. Thus privacy is not preserved for `low` with respect to the view definition.

In the next section, we present the syntax and the semantics of $\mathcal{ALC}$, explain how a query is answered on an $\mathcal{ALC}$ knowledge base, and recall from [3,4] the problem of provable data privacy on a given view. Then, in Section 3 we define data privacy on a view definition. We show that in order to decide this problem it is enough to consider a finite number of possible views. As a corollary we obtain that the problem is ExpTime-complete. Moreover, we present a syntactic condition on the knowledge base and the view definition which is sufficient for data privacy. This condition can be checked in PTime. We discuss related work in Section 4. Then we conclude and give some directions for further work.

This paper comes together with a technical report [5]. There we introduce a deductive system for $\mathcal{ALC}$ and apply proof-theoretic techniques in order to give detailed proofs of our results.

## 2 Preliminaries

The language of $\mathcal{ALC}$ consists of a countable set of *individuals* $\mathsf{Ind}$, a countable set of *atomic concepts* $\mathsf{AConc}$, a countable set of *roles* $\mathsf{Rol}$ and the *concepts* built on $\mathsf{AConc}$ and $\mathsf{Rol}$ 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. Individuals are denoted by $a, b, c, \ldots$.

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 $A \in \mathsf{AConc}$. The set of *subterms* $s(C)$ of a concept $C$ is defined by:

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

where $\star$ is either $\sqcup$ or $\sqcap$ and $Q$ is either $\forall$ or $\exists$. Note that the complements of atomic concepts are not decomposable. That means, 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 1.** *An interpretation $\mathcal{I}$ consists of a non-empty domain $\Delta^{\mathcal{I}}$ and a mapping $(\cdot)^{\mathcal{I}}$ 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:

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

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 $\top \sqsubseteq C$,[1] 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 $\psi$ an inclusion axiom or an assertion, we say that $\mathcal{O} \models \psi$ (in words, $\mathcal{O}$ *entails* $\psi$) if for every model $\mathcal{I}$ of $\mathcal{O}$, $\mathcal{I} \models \psi$ also holds.

The consistency problem for $\mathcal{ALC}$ is ExpTime-complete, see for instance [1]. The entailment problem is reducible to the consistency problem as follows:

**Theorem 1.** *Let $\mathcal{O}$ be an $\mathcal{ALC}$ knowledge base and $n_{ew}$ be an individual not belonging to* $\mathsf{Ind}(\mathcal{O})$*. 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 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.

**Definition 2.** *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 an $\mathcal{ALC}$ knowledge base $\mathcal{O}$ (*$\mathsf{ans}(q, \mathcal{O})$*) is given as follows where* $\mathsf{tt}$ *is a special constant denoting 'true'.*

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

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

**Definition 3.** *A* view *$V_I$ of a view definition $V$ is a total function with domain $V$ such that if $\langle q, r \rangle \in V_I$, then*

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

*1.* $r \subseteq \mathsf{Ind}$ *and finite if* $q$ *is a retrieval query,*
*2.* $r \subseteq \{\mathit{tt}\}$ *if* $q$ *is a boolean query.*

*We say, that an* $\mathcal{ALC}$ *knowledge base* $\mathcal{O}$ entails *a view* $V_I$ *(*$\mathcal{O} \models V_I$*) if for each* $\langle q, r\rangle \in V_I$ *we have* $r = ans(q, \mathcal{O})$*.*

Note that a view can also be formulated as a set $A_{V_I}$ of axioms and assertions. We set

$$A_{V_I} := \{\top \sqsubseteq C \mid \langle \top \sqsubseteq C, \{\mathsf{tt}\}\rangle \in V_I\} \cup \\ \{a : C \mid \text{there is a set } \mathsf{In} \text{ with } \langle C, \mathsf{In}\rangle \in V_I \text{ and } a \in \mathsf{In}\}.$$

Our notion of a view entailed by a knowledge base relates to the standard notion of entailment as follows. Let $V_I$ be a view of a view definition $V$ such that $\mathcal{O} \models V_I$ for some $\mathcal{O}$. For each retrieval query $C$ in $V$ and all individuals $a$ we have $C(a) \in A_{V_I}$ iff $\mathcal{O} \models C(a)$. For each boolean query $\top \sqsubseteq C$ in $V$ we have $\top \sqsubseteq C \in A_{V_I}$ iff $\mathcal{O} \models \top \sqsubseteq C$.

We turn now to the problem of provable data privacy wrt. views. This problem has been examined for arbitrary data and knowledge bases in [3,4]. 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 $\mathcal{O}_{bg}$.

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 4.** *A knowledge base* $\mathcal{P}$ *is* possible *wrt.* $\langle \mathcal{O}_{bg}, V_I\rangle$ *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 $\mathsf{Poss}_{\langle \mathcal{O}_{bg}, V_I\rangle} \neq \emptyset$ which means that $\mathcal{O} \cup A_{V_I}$ is satisfiable.

**Definition 5.** *The* certain answers *to a query* $q$ *wrt.* $\langle \mathcal{O}_{bg}, V_I\rangle$ *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 6.** *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*

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

Note that this privacy notion is based on positive answers only. So we may have privacy for a query $C$ even when we know with certainty that some individual $a$ does not belong to $C$. The extreme case is when $\top \sqsubseteq \neg C$ is public knowledge. Then we know that $C$ must be empty and still we have privacy for $C$ (since there is no element for which we can infer that it belongs to $C$).

There are situations in which the certain answers to a query $q$ can be computed by issuing $q$ against a particular fixed data or knowledge base, see for instance [6,3]. In our setting, we simply can take $\mathcal{O}_{bg} \cup A_{V_I}$ for this purpose. Namely, we have

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

Therefore, we immediately get the following result.

**Theorem 2 (see [4, 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*

$$\mathsf{ans}(q, \mathcal{O}_{bg} \cup A_{V_I}) = \emptyset.$$

According to Definition 2, $\mathsf{ans}(q, \mathcal{O}_{bg} \cup A_{V_I})$ can be computed by a number of entailments which is linear the size of $\mathcal{O}_{bg} \cup A_{V_I}$. If $q$ is a retrieval query, then we need one entailment check for each individual occurring in $\mathcal{O}_{bg} \cup A_{V_I}$. If $q$ is a boolean query, then we trivially need only one entailment check. As it has already been stated, the entailment problem is reducible to the consistency problem which is solvable in ExpTime. Therefore, Theorem 2 provides an ExpTime decision procedure for the problem of data privacy on views.

The problem of $\mathcal{ALC}$ concept satisfiability wrt. a consistent TBox is also ExpTime-hard, see [7] and [1]. Note that the proof in [1] does not necessarily construct a *consistent* TBox, however an easy modification will do the job. Therefore, the $\mathcal{ALC}$ data privacy problem is ExpTime-complete since we have that 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 1.** *The problem of $\mathcal{ALC}$ data privacy for a query wrt. a view and a knowledge base is ExpTime-complete.*

## 3 Data Privacy on View Definitions

Let us now introduce the problem of provable data privacy wrt. view definitions. First, we introduce the following auxiliary notion.

**Definition 7.** *Let $\mathcal{O}_{bg}$ be an $\mathcal{ALC}$ knowledge base and $V$ be a view definition. A view $V_I$* is based on *$\langle \mathcal{O}_{bg}, V \rangle$ if it satisfies the following: (i) $V_I$ is a view of $V$ and (ii) $\mathsf{Poss}_{\langle \mathcal{O}_{bg}, V_I \rangle} \neq \emptyset$.*

**Definition 8.** *Let $\mathcal{O}_{bg}$ be an $\mathcal{ALC}$ knowledge base and $V$ be a view definition. Data privacy is preserved for $q$ wrt. $\langle \mathcal{O}_{bg}, V \rangle$ 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. $\langle \mathcal{O}_{bg}, V \rangle$.*

*Example 2.* We consider again the business information system storing information about key accounts, account managers, and their salaries. The general background knowledge states the following: An account manager gets a high or a low salary, see (1). If someone gets a high salary, then she handles key accounts only, see (2). The domain of the `handles` relation is the set of account managers, see (3). Formally, $\mathcal{O}_{bg}$ is the set of the following axioms:

$$\mathtt{account\_manager} = \mathtt{high} \sqcup \mathtt{low} \tag{1}$$

$$\mathtt{high} \sqsubseteq \forall \mathtt{handles.key\_account} \tag{2}$$

$$\exists \mathtt{handles}.\top = \mathtt{account\_manager} \tag{3}$$

Consider the view definition $V_1 := \{\exists \mathtt{handles.key\_account}\}$. Given this setting, the following two statements, for example, hold:

$$\text{privacy is preserved for } \mathtt{key\_account} \text{ with respect to } \langle \mathcal{O}_{bg}, V_1 \rangle \tag{$*$}$$

$$\text{privacy is preserved for } \mathtt{low} \text{ with respect to } \langle \mathcal{O}_{bg}, V_1 \rangle. \tag{$**$}$$

That means an agent who is granted access to the view provided by $V_1$ cannot infer which are the key accounts nor who gets a low salary.

To see ($*$), simply observe that the only information we obtain from a non-empty answer to the query in $V_1$ is that the extension of `key_account` cannot be empty. However, we do not get any knowledge about which individual belongs to it. There is a possible knowledge base in which only some individual $a$ belongs to `key_account` and there is another possible knowledge base in which only some other individual $b$ belongs to `key_account`. Therefore the set of certain answers to `key_account` is empty and thus privacy is preserved.

To see ($**$), simply observe that we always can choose `low` to be the empty concept, no matter what the answer to the query in $V_1$ is.

Consider now the view definition

$$V_2 := \{\exists \mathtt{handles}.\neg \mathtt{key\_account}\}.$$

Privacy is not preserved for `low` with respect to $\langle \mathcal{O}_{bg}, V_2 \rangle$. This can be seen as follows. Assume that issuing the view query against some knowledge base $\mathcal{KB}$ gives

$$a \in \mathsf{ans}(\exists \mathtt{handles}.\neg \mathtt{key\_account}, \mathcal{KB}),$$

for some individual $a$. By (2), we get $\mathcal{KB} \models a : \neg \mathtt{high}$ and by (3) we find $\mathcal{KB} \models a : \mathtt{account\_manager}$. Thus (1) yields $\mathcal{KB} \models a : \mathtt{low}$. We conclude that privacy is not preserved for `low`.

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 $\langle \mathcal{O}_{bg}, V \rangle$ 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.** *Let $\mathcal{O}$ be an $\mathcal{ALC}$ knowledge base and $V$ be a view definition. Data privacy is preserved for $q$ wrt. $\langle \mathcal{O}_{bg}, V \rangle$ 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. $\langle \mathcal{O}_{bg}, V_I \rangle$.*

A proof is presented in [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 1 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. $\mathcal{O}_{bg} \cup A_{V_I}$ and the empty view definition.

**Theorem 4.** *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 9.** *A* query $q$ is trivial wrt. a tuple $\langle \mathcal{O}_{bg}, V \rangle$ *when*

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

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 $q^{\mathrm{C}}$ where $C$ is the inclusion or assertional concept of $q$ (i.e. $q^{\mathrm{C}} = \top \sqsubseteq C$ or $q^{\mathrm{C}} = C$). The function $s_{afe}()$ is defined on concepts and roles as follows:

For a concept $D$, $s_{afe}(D, q^{\mathrm{C}}) = 1$ 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 $A \in \mathsf{AConc}$, $R \in \mathsf{Rol}$ 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 $\langle \mathcal{O}_{bg}, V\rangle$, $s_{afe}(R, \langle \mathcal{O}_{bg}, V, q^{\mathrm{C}}\rangle) = 1$ iff:

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

The following theorem provides a sufficient condition for privacy on view definition. A proof can be found in our technical report [5]. There, the theorem is established by proof-theoretic investigations of a sequent system for $\mathcal{ALC}$.

**Theorem 5.** *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 $\langle \mathcal{O}_{bg}, V\rangle$*

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

*Moreover, it can be decided in PTime whether for every concept $D$ and role $R$ of $\langle \mathcal{O}_{bg}, V\rangle$ we have $s_{afe}(D, q) = s_{afe}(R, \langle \mathcal{O}_{bg}, V, q\rangle) = 1$.*

*Example 3.* Consider again the setting of Example 2. We can establish (∗) by the previous theorem. Let $q$ be the query `key_account`. We have

$$\begin{aligned} s_{afe}(\texttt{account\_manager}, q) &= s_{afe}(\texttt{high} \sqcup \texttt{low}, q) = s_{afe}(\texttt{high}, q) \\ &= s_{afe}(\exists\texttt{handles}.\top, q) = 1. \end{aligned}$$

We also have

$$s_{afe}(\forall\texttt{handles.key\_account}, q) = 1$$
$$s_{afe}(\exists\texttt{handles.key\_account}, q) = 1$$

since `key_account` occurs only behind a quantifier in the concepts of $\mathcal{O}_{bg}$ and $V_1$. Therefore the condition of Theorem 5 is satisfied and thus privacy is preserved for `key_account`.

However, Theorem 5 does not yield (∗∗) since `low` is an atomic subterm of $\texttt{high} \sqcup \texttt{low}$ which is not behind a quantifier.

## 4 Related Work

The notion of certain answer originates from the study of incomplete databases [8] and is now a key notion in data integration [9,10] and data exchange [11,6]. Obviously, our work on privacy in ontologies is tightly related to privacy in incomplete databases which has been studied by several authors.

Nash and Deutsch [12], for instance, study privacy for database integration. Like us, they are interested in logical security. That is all an attacker can do is issue queries and apply arbitrary computational power on the answers to these queries together with background knowledge to obtain the secret. They introduce several notions of privacy that are suitable for a data integration scenario and study the corresponding algorithms.

Another approach is to preserve confidentially at runtime. At each query, it is checked whether the answer would leak hidden information. If this is the case, then the answer is distorted. Biskup and Weibert [13] adopt this approach for incomplete databases. In their setting a database is simply a set of propositional sentences and queries simply return *yes, no,* or *undef.* They investigate several distortion methods (lying, refusal and a combination thereof) which guarantee that a user cannot learn classified information.

Our notion of provable data privacy only guarantees that, given a concept $C$, for no individual $a$ we can infer $a : C$. Example 2 shows that (∗) holds even if we know that `key_account` cannot be empty. *Perfect privacy* is a much more restrictive notion than provable privacy. It guarantees that an answer to a query does not change the attacker's a priori belief about the secret. This belief is modeled as a probability distribution with the assumption that the tuples in the secret answer are independent events. Perfect privacy has been introduced in [14] and generalized in [15]. Recently, a connection between perfect privacy and query containment has been established [16] which allows to identify subclasses of conjunctive queries for which enforcing perfect privacy is tractable. Dalvi et al. [17]

argue that perfect privacy is often too restrictive for practical applications. They provide a new probabilistic database model for practical privacy and study five privacy characterizations for it, including perfect privacy and certain answers.

It is necessary to consider an attacker's background knowledge when reasoning about privacy. We have chosen a simple approach: we fix the background knowledge and model it as a part of the knowledge base. The general case is when the background knowledge is not given in advance. Recently, a formal study of this so-called worst-case background knowledge has been initiated [18].

The problem of privacy aware access to ontologies is also addressed in [19]. There it is shown how view based query answering is able to conceal from the user information that are not logical consequences of the associated authorization views. The authors introduce several different semantics for view based query answering which in turn conceal different amounts of information when applied to the privacy problem. The semantics which corresponds to our approach is called TBox-centered semantics. There the user is aware of the TBox which in our setting is expressed by the TBox being part of the general background knowledge.

Grau and Horrocks [20] study different privacy guarantees for logic-based information systems. They present privacy preserving query answering as reasoning problems and establish a general connection between such reasoning problems and probabilistic privacy guarantees. The reasoning problems they introduce are related to certain notions of conservative extension which occur in the context of modular ontologies.

## 5 Conclusions

We have studied the problem of provable data privacy on view definitions for $\mathcal{ALC}$ knowledge bases. Our goal was to verify that a given privacy condition holds on all possible views of a given definition. We have presented an ExpTime-complete decision procedure for this privacy problem. Moreover, we have studied a syntactic condition which is sufficient for provable privacy and which can be decided in PTime.

Our work is preliminary in the sense that we treat only the case of $\mathcal{ALC}$ knowledge bases and views that are simple queries. There are two important generalizations of our results which will be addressed in future work.

First we have only considered $\mathcal{ALC}$ knowledge bases. $\mathcal{ALC}$ is the basic description logic language and therefore a natural candidate for an initial study. However, current ontology languages are based on very expressive description logics. Future work has to deal with, for instance, $\mathcal{SHOIN}(D)$ [21] which corresponds to OWL DL.

Second we restricted ourselves to concept retrieval and subsumption queries. In this setting, Theorem 3 becomes a consequence of the tree model property. Things are more complex if we also allow role expressions in a view. In a general setting, one also has to consider (union) conjunctive queries over description logics [22]. Then we cannot encode views as knowledge bases, and computing certain answers becomes more difficult.

## Acknowledgments

We would like to thank the anonymous referees for the detailed comments which helped to improve the quality of this paper.

## References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook. Cambridge University Press, Cambridge (2003)
2. Smith, M.K., Welty, C., McGuinness, D.L.: OWL web ontology language guide (2004), http://www.w3.org/TR/owl-guide/
3. Stoffel, K., Studer, T.: Provable data privacy. In: Viborg, K., Debenham, J., Wagner, R. (eds.) DEXA 2005. LNCS, vol. 3588, pp. 324–332. Springer, Heidelberg (2005)
4. Stouppa, P., Studer, T.: A formal model of data privacy. In: Virbitskaite, I., Voronkov, A. (eds.) PSI 2006. LNCS, vol. 4378, pp. 401–411. Springer, Heidelberg (2007)
5. Stouppa, P., Studer, T.: Data privacy for $\mathcal{ALC}$, Technical Report, IAM 08-002 (2008), http://www.iam.unibe.ch/publikationen/techreports/2008/
6. Fagin, R., Kolaitis, P.G., Miller, R., Popa, L.: Data exchange: Semantics and query answering. Theoretical Computer Science 336, 89–124 (2005)
7. Hofmann, M.: Proof-theoretic approach to description-logic. In: LICS 2005, pp. 229–237. IEEE Computer Society, Los Alamitos (2005)
8. van der Meyden, R.: Logical approaches to incomplete information: a survey. In: Logics for databases and information systems, pp. 307–356. Kluwer, Dordrecht (1998)
9. Calì, A., Calvanese, D., Giacomo, G.D., Lenzerini, M.: Data integration under integrity constraints. In: Pidduck, A.B., Mylopoulos, J., Woo, C.C., Ozsu, M.T. (eds.) CAiSE 2002. LNCS, vol. 2348, pp. 262–279. Springer, Heidelberg (2002)
10. Halevy, A.Y.: Answering queries using views: A survey. The VLDB Journal 10(4), 270–294 (2001)
11. Arenas, M., Libkin, L.: XML data exchange: Consistency and query answering. In: PODS, pp. 13–24 (2005)
12. Nash, A., Deutsch, A.: Privacy in glav information integration. In: Schwentick, T., Suciu, D. (eds.) ICDT 2007. LNCS, vol. 4353, pp. 89–103. Springer, Heidelberg (2006)
13. Biskup, J., Weibert, T.: Keeping secrets in incomplete databases. Journal of Information Security 7(3), 199–217 (2008)
14. Miklau, G., Suciu, D.: A formal analysis of information disclosure in data exchange. In: SIGMOD, pp. 575–586. ACM, New York (2004)
15. Deutsch, A., Papakonstantinou, Y.: Privacy in database publishing. In: Eiter, T., Libkin, L. (eds.) ICDT 2005. LNCS, vol. 3363, pp. 230–245. Springer, Heidelberg (2004)
16. Machanavajjhala, A., Gehrke, J.: On the efficiency of checking perfect privacy. In: PODS 2006, pp. 163–172. ACM Press, New York (2006)
17. Dalvi, N.N., Miklau, G., Suciu, D.: Asymptotic conditional probabilities for conjunctive queries. In: Eiter, T., Libkin, L. (eds.) ICDT 2005. LNCS, vol. 3363, pp. 289–305. Springer, Heidelberg (2004)

18. Martin, D.J., Kifer, D., Machanavajjhala, A., Gehrke, J., Halpern, J.Y.: Worst-case background knowledge in privacy. In: ICDE, pp. 126–135. IEEE, Los Alamitos (2007)
19. Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: View-based query answering over description logic ontologies. In: Principles of Knowledge Representation and Reasoning, pp. 242–251 (2008)
20. Grau, B.C., Horrocks, I.: Privacy-preserving query answering in logic-based information systems. In: 18th European Conference on Artificial Intelligence (ECAI 2008) (to appear, 2008)
21. Horrocks, I., Patel-Schneider, P., van Harmelen, F.: From SHIQ and RDF to OWL: The making of a web ontology language. Journal of Web Semantics 1(1) (2003)
22. Calvanese, D., Lenzerini, M.: Answering queries using views over description logics knowledge bases. In: Proceedings of AAAI 2000, pp. 386–391 (2000)