Secrecy-preserving Query Answering for Instance Checking in $\mathcal{EL}$

Jia Tao, Giora Slutzki, and Vasant Honavar

Iowa State University, Ames, IA, USA

Abstract. We consider the problem of answering queries against an $\mathcal{EL}$ knowledge base (KB) using secrets, whenever it is possible to do so without compromising secrets. We provide a polynomial time algorithm that, given an $\mathcal{EL}$ KB Σ, a set $\mathbb{S}$ of secrets to be protected and a query q, outputs "Yes" whenever $\Sigma \models q$ and the answer to q, together with the answers to any previous queries answered by the KB, does not allow the querying agent to deduce any of the secrets in $\mathbb{S}$. This approach allows more flexible information sharing than is possible with traditional access control mechanisms.

1 Introduction

The rapid expansion of the World Wide Web and the widespread use of distributed databases and networked information systems offer unprecedented opportunities for productive interaction and collaboration among individuals and organizations in virtually every area of human endeavor. However, the need to share information has to be balanced against the need to protect secrets. Such scenarios call for algorithms that can, given a knowledge base Σ and a set $\mathbb{S}$ of secrets (perhaps specified using some secrecy policy), answer queries against Σ, using secrets if necessary, whenever it is possible to do so without compromising secrets (See Example 1 in Sect. 2). Most existing approaches to information protection simply *forbid* the use of secret information in answering queries (See Sect. 4). The *privacy-preserving reasoning* framework introduced in [1] was motivated by the need to alleviate, at least in part, this limitation in the simple setting of *hierarchical* knowledge bases (KBs) under the *open world assumption* (OWA)[1]. Such KBs may contain scientific, medical, economic information, or military intelligence, etc. Our secrecy-preserving reasoning framework builds on, and substantially extends, the framework introduced by Bao et al. [1].

In general, the answer to a query q of the form $C(a)$ or $r(a, b)$ against a KB Σ can be "Yes" (i.e., q can be inferred from Σ), "No" ($\neg q$ can be inferred from Σ) or "Unknown" (e.g., because of the incompleteness of Σ). We assume cooperative as opposed to adversarial scenarios where the KB does not *lie*. However,

[1] Under the closed world assumption a statement that cannot be inferred from the KB to be true, is presumed to be false. Under the OWA, the truth of such a statement is presumed to be unknown, and *not necessarily* false.

whenever truthfully answering a query risks compromising secrets in $\mathbb{S}$, the reasoner associated with the KB is allowed to feign ignorance, i.e., answer the query as "Unknown". Given a set of secrets $\mathbb{S}$ (which need not be a subset of Σ), it is clear that, to protect $\mathbb{S}$, answers to queries in $\mathbb{S}$ will be "Unknown". However, in general, it is not sufficient to protect only $\mathbb{S}$ since truthful answers to certain queries (not in $\mathbb{S}$) may reveal information in $\mathbb{S}$. Therefore, we must protect a superset of $\mathbb{S}$, which we call an *envelope* of $\mathbb{S}$, such that the querying agent who has no access to the envelope will not be able to deduce any information in $\mathbb{S}$.

In this paper, we investigate secrecy-preserving query answering with $\mathcal{EL}$ [2], which is one of the simplest DLs that is both computationally tractable [3, 4] and practically useful [2]. For example, the medical ontology SNOMED CT [5] and large parts of the medical ontology GALEN [6] can be expressed in $\mathcal{EL}$. We provide algorithms to answer queries against an $\mathcal{EL}$ KB that use, but not reveal, the information that is designated as secret. Because of the open world assumption and the fact that the language of $\mathcal{EL}$ does not include negation, the answer to a query can only be "Yes" or "Unknown".

To answer queries posed to the KB, we construct a *secrecy maintenance system* that consists of: a finite set of consequences of the KB Σ, denoted by $\mathcal{A}^*$, and a secrecy envelope $\mathbb{S} \subseteq \mathbb{E}_{\mathbb{S}} \subseteq \mathcal{A}^*$. The answer to a query q is censored by the reasoner if $q \in \mathbb{E}_{\mathbb{S}}$. It is easy to see that a secrecy envelope always exists. For instance, $\mathcal{A}^*$ constitutes an envelope for any secrecy set $\mathbb{S} \subseteq \mathcal{A}^*$. A key challenge is to *develop strategies that can be used by the KB to respond to queries as informatively as possible (i.e., using an envelope that is as small as possible) without compromising secrets that the KB is obliged to protect.* Unfortunately, computing a minimum envelope is NP-hard [7]. We compute $\mathcal{A}^*$ using the (usual) tableau expansion rules. To compute $\mathbb{E}_{\mathbb{S}}$, we introduce the following idea. From each original expansion rule, we construct a corresponding *inverse expansion rule*. We show that the inverted system of expansion rules generates an envelope of $\mathbb{S}$. To the best of our knowledge, the idea of constructing a secrecy envelope by inverting the tableau expansion rules is novel. Furthermore, we introduce a couple of useful optimizations that help reduce the size of an envelope.

2 Preliminaries

The non-logical signature of the $\mathcal{EL}$ description language includes three mutually disjoint sets: *concept names* $N_{\mathcal{C}}$, *role names* $N_{\mathcal{R}}$ and *individual names* $N_{\mathcal{O}}$. The syntax of $\mathcal{EL}$ is defined by specifying *expressions* and *formulae*. $\mathcal{EL}$ expressions consist of $N_{\mathcal{R}}$ and the set of *concepts* $\mathcal{C}$ recursively defined as follows:

$$C, D \longrightarrow A \mid \top \mid C \sqcap D \mid \exists r.C$$

where $A \in N_{\mathcal{C}}$, $\top$ is the *top symbol*, $C, D \in \mathcal{C}$ and $r \in N_{\mathcal{R}}$. In this paper we consider three kinds of $\mathcal{EL}$ formulae: *assertions* of the form $C(a)$ or $r(a, b)$, *definitions* of the form $A \doteq D$ and *general concept inclusions (GCI)* of the form $C \sqsubseteq D$ where $a, b \in N_{\mathcal{O}}$, $C, D \in \mathcal{C}$, $r \in N_{\mathcal{R}}$ and $A \in N_{\mathcal{C}}$.

The semantics of $\mathcal{EL}$ is specified by means of an *interpretation* $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ where Δ is a non-empty domain and $\cdot^{\mathcal{I}}$ is a function that maps each individual name to an element in Δ, each concept name to a subset of Δ and each role name to a subset of $\Delta \times \Delta$. The interpretation of concept expressions is extended recursively: for $r \in N_{\mathcal{R}}$ and $C, D \in \mathcal{C}$: $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$ and $(\exists r.C)^{\mathcal{I}} = \{a \in \Delta \mid \exists b \in \Delta : (a, b) \in r^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$.

A finite non-empty set of assertions is called an *ABox*. A finite set of definitions and GCIs is called a *TBox*. An ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ whose concepts and roles belong to the language $\mathcal{EL}$ form an $\mathcal{EL}$-knowledge base $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$. A TBox $\mathcal{T}$ is *normalized* [3] if $\mathcal{T}$ contains only GCIs all of which are of one of the following forms: $A \sqsubseteq B$, $A_1 \sqcap A_2 \sqsubseteq B$, $A \sqsubseteq \exists r.B$ or $\exists r.A \sqsubseteq B$ where $A, A_1, A_2, B \in N_{\mathcal{C}} \cup \{\top\}$. It was shown that transforming a TBox into such a normal form can be accomplished in polynomial time [3]. From now on, we will assume that all the TBoxes are in normal form. By N_{Σ} (resp. $\mathcal{O}_{\Sigma}$) we denote the set of all symbols (resp. individual names) occurring in Σ. Note that $\mathcal{O}_{\Sigma} \subset N_{\mathcal{O}} \cap N_{\Sigma}$ and $N_{\Sigma} \setminus \mathcal{O}_{\Sigma} \subset N_{\mathcal{C}} \cup N_{\mathcal{R}}$.

Definition 1. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$ be a knowledge base, $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ an interpretation, $C, D \in \mathcal{C}$, $r \in N_{\mathcal{R}}$ and $a, b \in N_{\mathcal{O}}$. $\mathcal{I}$* satisfies *$C(a)$, $r(a, b)$, or $C \sqsubseteq D$ if, respectively, $a^{\mathcal{I}} \in C^{\mathcal{I}}$, $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$, or $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. $\mathcal{I}$ is a* model *of Σ if it satisfies all the assertions in $\mathcal{A}$ and all the GCIs in $\mathcal{T}$. Let α be an assertion or a GCI. We say that Σ* entails *α, written as $\Sigma \models \alpha$, if all models of Σ satisfy α.*

Example 1. (a simplified version adapted from [8]) Given a KB $\Sigma_1 = \langle \mathcal{A}_1, \mathcal{T}_1 \rangle$ that contains information on the patients, their health history, the prescriptions that they get from the physicians and their insurance information. Suppose that Jane*'s* mother Jill had breast cancer and that Jane tests positive for BRCA1 mutation which is linked to an increased risk of breast cancer. To reduce the breast cancer risk, Jane was prescribed a certain drug. Jane purchases the medications from her pharmacy and wants to get reimbursed for the cost of her prescription by her insurance company. If her insurance company finds out that she has tested positive for BRCA1 mutation or that she has been prescribed certain drug(s) for breast cancer, Jane risks losing her health insurance. The scenario can be formally specified in the DL $\mathcal{EL}$ as follows:

1. $\exists$is_child.A $\sqsubseteq$ CancerRisk
2. HasMutBRCA1 $\sqsubseteq$ $\exists$has_pres.CancerDrug
3. $\exists$has_pres.CancerDrug $\sqsubseteq$ CancerRisk
4. $\exists$has_pres.CoveredDrug $\sqsubseteq$ Reimburse
5. CancerDrug $\sqsubseteq$ CoveredDrug
6. A $\sqsubseteq$ Woman
7. A $\sqsubseteq$ HasCancer
8. Woman $\sqcap$ HasCancer $\sqsubseteq$ A
9. Woman(Jill)
10. HasCancer(Jill)
11. is_child(Jane, Jill)
12. HasMutBRCA1(Jane)

The GCIs 1-8 form a subset of $\mathcal{T}_1$ (in normal form) and the assertions 9-12 form a subset of $\mathcal{A}_1$. In order for Jane to get reimbursed, when the query Reimburse(Jane) is posed to the KB, the answer should be "Yes". However, in order to protect Jane's privacy, the query CancerRisk(Jane) should be answered "Unknown". ■

3 The Secrecy-preserving Query Answering

Problem Statement: Given a knowledge base Σ and a finite secrecy set $\mathbb{S}$, the basic goal is to answer queries while preserving secrecy. As shown in Example 1, to protect Jane's privacy, the query CancerRisk(Jane) should be answered "Unknown". However, by only keeping CancerRisk(Jane) secret, the fact that Jane has cancer risk can still be inferred by statements 12, 2 and 3. Therefore, the secrecy-preserving query answering problem is to find a superset of $\mathbb{S}$, which we call the *secrecy envelope* of $\mathbb{S}$, denoted by $\mathbb{E}_{\mathbb{S}}$, so that by protecting $\mathbb{E}_{\mathbb{S}}$, the querying agent cannot conclude anything in $\mathbb{S}$. Because of the OWA, when the answer to a query is "Unknown", the querying agent is not able to distinguish between (a) the answer to the query is truly unknown, or (b) the answer is being protected for reasons of secrecy.

The framework contains following components. We assume a KB $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$, a reasoner $\mathfrak{R}$ that is complete, and a secrecy set $\mathbb{S}$ consisting of a finite set of assertions that contain only symbols from N_{Σ}. $\mathfrak{R}$ is used to answer queries by checking whether the query can be inferred from Σ and if it can, whether answering "Yes" will reveal secrets from $\mathbb{S}$. The specific tasks are:

- To compute the set *SubC* of sub-expressions of all concepts and roles appearing in Σ or $\mathbb{S}$.
- To compute the set of all assertional consequences of Σ restricted to *SubC*. This set is called the *assertional closure of* Σ and it is denoted by $\mathcal{A}^*$. We assume that $\mathbb{S} \subseteq \mathcal{A}^*$.
- To compute the secrecy envelope $\mathbb{S} \subseteq \mathbb{E}_{\mathbb{S}} \subseteq \mathcal{A}^*$, a set of assertions which if truthfully answered, may reveal some secret(s) in $\mathbb{S}$.
- To answer queries. If a query cannot be inferred from Σ, the answer is simply "Unknown". If it can be inferred and it is not in $\mathbb{E}_{\mathbb{S}}$, the answer is "Yes"; otherwise, the answer is "Unknown".

We also assume that the querying agent (i) asks queries of the form $C(a)$ or $r(a, b)$; (ii) has computational access only to the signature of the knowledge base, i.e., its queries are over N_{Σ}; and (iii) has the same reasoning capacity as $\mathfrak{R}$ (Since we assume that $\mathfrak{R}$ is complete, this is not a restriction.); (iv) may log the history of all the answers to its queries and draw conclusions from it; and (v) has access to the TBox $\mathcal{T}$.

$\mathcal{A}^*$ and $\mathbb{E}_{\mathbb{S}}$ form a *secrecy maintenance system.* Note that the both are restricted to *SubC*. Once $\mathcal{A}^*$ and $\mathbb{E}_{\mathbb{S}}$ have been computed, if $C \in SubC$, $\mathfrak{R}$ can answer the query $C(a)$ in linear time depending on its membership of $\mathcal{A}^*$ and $\mathbb{E}_{\mathbb{S}}$. Otherwise, we need to expand *SubC* by adding sub-expressions of C that are not in *SubC* and update the consequences $\mathcal{A}^*$ as well as $\mathbb{E}_{\mathbb{S}}$ accordingly.

3.1 Initializing Secrecy Maintenance System

Computing *SubC*: The set of certain sub-expressions of all the concepts and roles appearing in Σ or $\mathbb{S}$, is defined as follows:

if $C(a) \in \mathcal{A} \cup \mathbb{S}$, then $C \in \mathit{Sub}\mathcal{C}$; if $C \sqsubseteq D \in \mathcal{T}$, then $\{C, D\} \subseteq \mathit{Sub}\mathcal{C}$;
if $r(a, b) \in \mathcal{A} \cup \mathbb{S}$, then $r \in \mathit{Sub}\mathcal{C}$; if $\exists r.C \in \mathit{Sub}\mathcal{C}$, then $\{r, C\} \subseteq \mathit{Sub}\mathcal{C}$;
if $C_1 \sqcap \cdots \sqcap C_k \in \mathit{Sub}\mathcal{C}(C_i \in N_{\mathcal{C}}$ or $C_i = \exists r.C)$, then $C_i \in \mathit{Sub}\mathcal{C}(1 \leq i \leq k)$;
if $\exists r.C \in \mathit{Sub}\mathcal{C}$ and $C \sqsubseteq D \in \mathcal{T}$ or $D \sqsubseteq C \in \mathcal{T}$, then $\exists r.D \in \mathit{Sub}\mathcal{C}$.

Note that $\mathit{Sub}\mathcal{C}$ does not contain all the sub-expressions of concepts appearing in Σ or $\mathbb{S}$. If a query $C(a)$ comes along where $C \notin \mathit{Sub}\mathcal{C}$, it will be added into $\mathit{Sub}\mathcal{C}$. As such, the secrecy maintenance system is built up gradually depending on the history of queries. Also note that the initial size of $\mathit{Sub}\mathcal{C}$ is linear in the size of the knowledge base Σ plus the size of the secrecy set $\mathbb{S}$.

Computing $\mathcal{A}^*$: The ABox $\mathcal{A}^*$ is initialized as $\mathcal{A}$ and expanded by recursively applying assertion expansion rules listed in Fig. 1. We say that $\mathcal{A}^*$ is *assertionally closed* or that it is an *assertional closure of* Σ if no assertion expansion rule is applicable. The set of all the individual names appearing in $\mathcal{A}^*$ is denoted by $\mathcal{O}^*$. It is initialized as $\mathcal{O}_\Sigma$ and is expanded with applications of the $\exists_2^{\mathcal{A}}$-rule. An individual a is said to be *fresh* (at a particular time during the expansion process) if $a \in N_{\mathcal{O}} \setminus \mathcal{O}^*$ (at that time). An individual $a \in \mathcal{O}^*$ is *blocked* by an individual $b \in \mathcal{O}^*$ if $a \in \mathcal{O}^* \setminus \mathcal{O}_\Sigma$, b is either in $\mathcal{O}_\Sigma$ or b was picked earlier than a (during the expansion process), and $\{C \mid C(a) \in \mathcal{A}^*\} \subseteq \{C' \mid C'(b) \in \mathcal{A}^*\}$. Recall that we have assumed that the querying agent has computational access only to the signature of the knowledge base. In particular, the querying agent cannot ask any queries that involve individual names in $\mathcal{O}^* \setminus \mathcal{O}_\Sigma$. This is referred to as *Hidden Names Assumption* (HNA).

$\sqcap_1^{\mathcal{A}}$ -rule: if $C_1 \sqcap \cdots \sqcap C_k(a) \in \mathcal{A}^*$ and $C_i(a) \notin \mathcal{A}^*$,
then $\mathcal{A}^* := \mathcal{A}^* \cup \{C_i(a)\}$ where $1 \leq i \leq k$;
$\sqcap_2^{\mathcal{A}}$ -rule: if $\{C_1(a), ..., C_k(a)\} \subseteq \mathcal{A}^*, C_1 \sqcap \cdots \sqcap C_k \in \mathit{Sub}\mathcal{C}$
and $C_1 \sqcap \cdots \sqcap C_k(a) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{C_1 \sqcap \cdots \sqcap C_k(a)\}$;
$\exists_1^{\mathcal{A}}$ -rule: if $\{r(a, b), C(b)\} \subseteq \mathcal{A}^*, \exists r.C \in \mathit{Sub}\mathcal{C}$ and $\exists r.C(a) \notin \mathcal{A}^*$,
then $\mathcal{A}^* := \mathcal{A}^* \cup \{\exists r.C(a)\}$;
$\exists_2^{\mathcal{A}}$ -rule: if $\exists r.C(a) \in \mathcal{A}^*, a$ is not blocked and $\forall b \in \mathcal{O}^*, \{r(a, b), C(b)\} \nsubseteq \mathcal{A}^*$,
then $\mathcal{A}^* := \mathcal{A}^* \cup \{r(a, c), C(c)\}$ where c is fresh, and $\mathcal{O}^* := \mathcal{O}^* \cup \{c\}$;
$\sqsubseteq^{\mathcal{T}}$ -rule: if $C(a) \in \mathcal{A}^*, C \sqsubseteq D \in \mathcal{T}$ and $D(a) \notin \mathcal{A}^*$, then $\mathcal{A}^* := \mathcal{A}^* \cup \{D(a)\}$;

Fig. 1. Assertion Expansion Rules

We denote by Λ the tableau algorithm which nondeterministically applies assertion expansion rules until no further applications are possible. Since each expansion rule can be applied polynomially many times (in the size of $\mathit{Sub}\mathcal{C}$), the computation of $\mathcal{A}^*$ can be done in polynomial time. When an execution of Λ terminates, we have an assertionally closed ABox $\mathcal{A}^*$. The soundness and completeness of the Λ-*tableau algorithm* are proved in [7].

Ignoring the issue of secrecy, we point out a difference between the reasoning of the KB reasoner $\mathfrak{R}$ and that of the querying agent. Consider the assertion $\exists r.C(a) \in \mathcal{A}^*$ when a is not blocked and there does not exist $b \in \mathcal{O}_\Sigma$ for which $\{r(a,b), C(b)\} \subseteq \mathcal{A}^*$. In this case $\mathfrak{R}$ picks a fresh individual name $c \notin \mathcal{O}_\Sigma$ as a witness for the inclusion $\exists r.C(a) \in \mathcal{A}^*$. The querying agent only knows the existence of the witness individual and not the individual name itself. Of course, for its own reasoning process, the querying agent may pick any individual name in $N_\mathcal{O} \setminus \mathcal{O}_\Sigma$, say d, and then force $r(a,d)$ and $C(d)$ to be consequences of Σ. Clearly, the reasoner $\mathfrak{R}$ and the querying agent are not aware of each other's "fresh" individual names. To differentiate the assertional closure of the KB reasoner $\mathfrak{R}$ from the reasoning of the querying agent, we will use $\cdot^+$ to denote the latter.

Computing the Secrecy Envelopes: We define the secrecy envelope $\mathbb{E}_\mathbb{S}$ such that if the reasoner $\mathfrak{R}$ answers every query in $\mathbb{E}_\mathbb{S}$ with "Unknown" and every query in $\mathcal{A}^* \setminus \mathbb{E}_\mathbb{S}$ with "Yes", the querying agent will not be able to deduce any assertions in $\mathbb{S}$.

Definition 2. *Given a knowledge base* $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$ *and a finite secrecy set* $\mathbb{S} \subseteq \mathcal{A}^*$*, a* secrecy envelope *of* $\mathbb{S}$*, denoted by* $\mathbb{E}_\mathbb{S}$*, is a superset* $\mathbb{S} \subseteq \mathbb{E}_\mathbb{S} \subseteq \mathcal{A}^*$ *such that* $(\mathcal{A}^* \setminus \mathbb{E}_\mathbb{S})^+ \cap \mathbb{S} = \emptyset$ *where* $(\mathcal{A}^* \setminus \mathbb{E}_\mathbb{S})^+$ *is the assertional closure of the knowledge base* $\langle \mathcal{A}^* \setminus \mathbb{E}_\mathbb{S}, \mathcal{T} \rangle$ *for the querying agent.*

To answer queries as informatively as possible, we aim to make $\mathbb{E}_\mathbb{S}$ as small as possible. Unfortunately, to compute a minimum envelope is hard. Specifically, the decision version of the problem of computing minimum envelopes is NP-complete [7]. In what follows, we provide an algorithm that computes envelopes. Utilizing the HNA, we further optimize the algorithm to result a smaller envelope. To compute an envelope, we introduce the novel idea of *inverting assertion expansion rules.* For $\mathcal{EL}$ with TBox, we have five assertion expansion rules as listed in Fig. 1. For each assertion expansion rule, the resulting inverse rule is named by changing the superscript in the name of the original rule to $\mathbb{S}$. These inversion rules are called $\mathfrak{R}$-*secrecy closure rules* and are listed in Fig. 2. In Fig. 2, $\mathcal{A}^*$ is assumed to have been computed previously; $\mathbb{E}$ is initialized to $\mathbb{S}$, and expanded by using $\mathfrak{R}$-secrecy closure rules.

$\sqcap_1^\mathbb{S}$ -rule: if $C_1 \sqcap \cdots \sqcap C_k(a) \in \mathcal{A}^* \setminus \mathbb{E}$ and $\{C_1(a), ..., C_k(a)\} \cap \mathbb{E} \neq \emptyset$,
then $\mathbb{E} := \mathbb{E} \cup \{C_1 \sqcap \cdots \sqcap C_k(a)\}$;
$\sqcap_2^\mathbb{S}$ -rule: if $C_1 \sqcap \cdots \sqcap C_k(a) \in \mathbb{E}$ and $\{C_1(a), ..., C_k(a)\} \cap \mathbb{E} = \emptyset$,
then $\mathbb{E} := \mathbb{E} \cup \{C_i(a)\}$ where $1 \leq i \leq k$;
$\exists_1^\mathbb{S}$ -rule: if $\exists r.C(a) \in \mathbb{E}$ and $\{r(a,b), C(b)\} \subseteq \mathcal{A}^* \setminus \mathbb{E}$ with $b \in \mathcal{O}^*$,
then $\mathbb{E} := \mathbb{E} \cup \{r(a,b)\}$ or $\mathbb{E} := \mathbb{E} \cup \{C(b)\}$;
$\exists_2^\mathbb{S}$ -rule: if $\exists r.C(a) \in \mathcal{A}^* \setminus \mathbb{E}$, and for every $b \in \mathcal{O}^*$ with $\{r(a,b), C(b)\} \subseteq \mathcal{A}^*$,
we have $\{r(a,b), C(b)\} \cap \mathbb{E} \neq \emptyset$, then $\mathbb{E} := \mathbb{E} \cup \{\exists r.C(a)\}$;
$\sqsubseteq^\mathbb{S}$ -rule: if $D(a) \in \mathbb{E}, C \sqsubseteq D \in \mathcal{T}$ and $C(a) \in \mathcal{A}^* \setminus \mathbb{E}$, then $\mathbb{E} := \mathbb{E} \cup \{C(a)\}$.

Fig. 2. $\mathfrak{R}$-secrecy closure rules obtained by inverting rules in Fig. 1.

We denote by $\Lambda_{\mathbb{S}}^{\mathfrak{R}}$ the tableau algorithm which nondeterministically applies the $\mathfrak{R}$-secrecy closure rules until no further rules are applicable. When no $\mathfrak{R}$-secrecy closure rule is applicable, we say that $\mathbb{E}$ is $\mathfrak{R}$-*closed*. It is easy to see that $\Lambda_{\mathbb{S}}^{\mathfrak{R}}$ terminates in polynomial time in the size of its input. The following lemma and corollary show that $\Lambda_{\mathbb{S}}^{\mathfrak{R}}$ results an envelope. The proofs are available in [7].

Lemma 1. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$ be a KB, $\mathbb{S} \subseteq \mathbb{E} \subseteq \mathcal{A}^*$ where $\mathbb{S}$ is the secrecy set and $\mathbb{E}$ is $\mathfrak{R}$-closed. Then (a) $\mathcal{A}^* \setminus \mathbb{E}$ is assertionally closed w.r.t. assertion expansion rules listed in Fig. 1, (b) $\mathbb{E}$ is a secrecy envelope of $\mathbb{S}$.*

It turns out that the $\Lambda_{\mathbb{S}}^{\mathfrak{R}}$ algorithm, although certainly producing an envelope, may actually result an envelope that is unnecessarily large. Specifically, even if $\exists_2^{\mathcal{A}}$-rule is applicable to $(\mathcal{A}^* \setminus \mathbb{E}_{\mathbb{S}})^+$, due to OWA, the querying agent can only conclude that there exists an individual d that is the witness for $\exists r.C(a)$ and that $d \notin \mathcal{O}_{\Sigma}$. However, by HNA, the querying agent has no computational access to individual names in $\mathcal{O}^* \setminus \mathcal{O}_{\Sigma}$. This provides a cue that when computing a secrecy envelope, the $\exists_2^{\mathbb{S}}$-rule, which inverts the $\exists_2^{\mathcal{A}}$-rule, is dispensable. The new set of secrecy closure rules, called $\mathcal{Q}$-*Secrecy Closure Rules*, includes only the $\sqcap_1^{\mathbb{S}}$-rule, the $\sqcap_2^{\mathbb{S}}$-rule, the $\sqsubseteq^{\mathbb{S}}$-rule and the $\exists_1^{\mathbb{S}}$-rule is replaced by an "optimized" version the $\exists^{\mathbb{S}}$-rule.

$\exists^{\mathbb{S}}$ -rule: if $\exists r.C(a) \in \mathbb{E}$ and $\{r(a,b), C(b)\} \subseteq \mathcal{A}^* \setminus \mathbb{E}$ with $b \in \mathcal{O}_{\Sigma}$,
then $\mathbb{E} := \mathbb{E} \cup \{r(a,b)\}$ or $\mathbb{E} := \mathbb{E} \cup \{C(b)\}$

We denote by $\Lambda_{\mathbb{S}}^{\mathcal{Q}}$ the tableau algorithm which nondeterministically and exhaustively applies the $\mathcal{Q}$-secrecy closure rules. The resulting $\mathbb{E}$ is said to be $\mathcal{Q}$-*closed*. It is clear that all executions of $\Lambda_{\mathbb{S}}^{\mathcal{Q}}$ terminate in polynomial time. Theorem 1 shows that $\Lambda_{\mathbb{S}}^{\mathcal{Q}}$ also results an envelope. Proofs can be found in [7].

Theorem 1. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$ be a KB, $\mathbb{S} \subseteq \mathbb{E} \subseteq \mathcal{A}^*$ where $\mathbb{S}$ is the secrecy set and $\mathbb{E}$ is $\mathcal{Q}$-closed. Then $\mathbb{E}$ is a secrecy envelope of $\mathbb{S}$.*

Note that the whole initialization of the secrecy maintenance system (including computation of $Sub\mathcal{C}$, $\mathcal{A}^*$ and $\mathbb{E}_{\mathbb{S}}$) is easily seen to be doable in polynomial time in the size of the KB Σ plus the size of the given secrecy set $\mathbb{S}$.

3.2 Query Answering

In this section we assume that the three sets $Sub\mathcal{C}$, $\mathcal{A}^*$ and $\mathbb{E}_{\mathbb{S}}$ (the latter two, restricted to $Sub\mathcal{C}$) have been precomputed in the pre-query stage as described in Sect. 3.1. The computation of the answer to a query of the form $C(a)$ is given in Fig. 3. The input of the secrecy-preserving query answering procedure SPQA contains the TBox $\mathcal{T}$ in normal form, precomputed assertional closure $\mathcal{A}^*$, the query $C(a)$ and the precomputed secrecy envelope $\mathbb{E}_{\mathbb{S}}$. Since sub-expressions of C, denoted by $sub(C)$, need not be in $Sub\mathcal{C}$, Line 2 in the SPQA procedure expands $Sub\mathcal{C}$ by adding expressions in $sub(C) \setminus Sub\mathcal{C}$. The expanded $Sub\mathcal{C}$ will be used to update $\mathcal{A}^*$ by applying assertion expansion rules (Fig. 1) until none of them is applicable, as indicated in Line 2. As a consequence, there may be applicable $\mathcal{Q}$-secrecy closure rules, implying that $\mathbb{E}_{\mathbb{S}}$ may no longer be a secrecy envelope for $\mathbb{S}$. Therefore, we apply necessary secrecy closure rules exhaustively (Line 3). Clearly, a single invocation of the procedure SPQA takes polynomial time (in the sum of the sizes of its arguments).

```
SPQA($\mathcal{T}, \mathcal{A}^*, C(a), \mathbb{E}_{\mathbb{S}}$):
1.  if ($C \notin SubC$) {
2.      compute $sub(C)$; $SubC = SubC \cup sub(C)$; expand $\mathcal{A}^*$ to $SubC$;
3.      expand the secrecy envelope $\mathbb{E}_{\mathbb{S}}$ to $SubC$; }
4.  if ($C(a) \in \mathcal{A}^*$ and $C(a) \notin \mathbb{E}_{\mathbb{S}}$) return "Yes";
5.  else return "Unknown";
```

Fig. 3. Secrecy-preserving Query-answering Procedure

For queries of the form $r(a,b)$, the procedure is much simpler: if $r(a,b) \in \mathcal{A} \setminus \mathbb{E}_{\mathbb{S}}$, then the answer is "Yes"; otherwise, the answer is "Unknown". Here $\mathbb{E}_{\mathbb{S}}$ is the current secrecy envelope.

Example 2. (Example 1, continued) Recall that we have a KB $\Sigma_1 = \langle \mathcal{A}_1, \mathcal{T}_1 \rangle$ and the secrecy set $\mathbb{S}_1 = \{\text{CancerRisk(Jane)}\}$. The assertional closure of Σ_1, denoted by $\mathcal{A}_1^*$, and one possible envelope $\mathbb{E}_{\mathbb{S}1}$ are listed below:

$\mathcal{A}_1^* = \mathcal{A}_1 \cup \{$ A(Jill), $\exists$is_child.A(Jane), CancerRisk(Jane), has_pres(Jane, a),
$\exists$has_pres.CancerDrug(Jane), CancerDrug(a), CoveredDrug(a),
$\exists$has_pres.CoveredDrug(Jane), Reimburse(Jane)$\}$.

$\mathbb{E}_{\mathbb{S}1} = \{$CancerRisk(Jane), is_child(Jane, Jill), HasMutBRCA1(Jane),
$\exists$is_child.A(Jane), $\exists$has_pres.CancerDrug(Jane)$\}$.

If the querying agent asks the query Reimburse(Jane), Reimburse(Jane)$\in \mathcal{A}_1^* \setminus \mathbb{E}_{\mathbb{S}1}$, the answer to the query is "Yes". If the querying agent asks the query CancerRisk(Jane), since CancerRisk(Jane)$\in \mathcal{A}_1^* \cap \mathbb{E}_{\mathbb{S}1}$, the answer to the query is "Unknown". ■

4 Summary and Discussion

Summary: In this paper, we have introduced a logic-based framework for secrecy preserving query answering in $\mathcal{EL}$ knowledge bases. We have provided a polynomial time algorithm that, given an $\mathcal{EL}$ KB Σ, a set $\mathbb{S}$ of secrets to be protected and a query q, truthfully answers the query whenever: (i) $\Sigma \models q$ and (ii) the answer to q, together with the answers to any previous queries answered by the KB does not allow the querying agent to deduce any of the secrets in $\mathbb{S}$. Our approach exploits the open world semantics under which it is impossible for the querying agent to distinguish between an answer "Unknown" resulting because of incomplete knowledge of the KB or because of selective censoring of answers by the KB. Our secrecy-preserving reasoning framework builds on, and substantially extends, the privacy-preserving reasoning framework introduced by Bao et al. [1] which considered protecting class-subclass relationships in hierarchical ontologies.

Related Work: Most of the work in this area falls into four broad categories of access control mechanisms, information confinement, preventing disclosure of information of specific individuals and controlled query evaluation. In contrast, our approach permits the use of secrets in answering queries for a given KB when it is possible to do so without compromising secrets under the OWA. A detailed comparison can be found in [7].

Future Work: Some natural directions for future work include: (i) design of an efficient algorithm for computing a "tight" envelope for $\mathcal{EL}$ KBs, i.e., an envelope from which no statement can be dropped without risking the possibility of secrets being compromised (such an algorithm is of interest in light of the fact that our current algorithm is not guaranteed to produce a tight envelope and the fact that computing the minimum envelope is NP-hard); (ii) exploration of secrecy-preserving query answering algorithms in the case of more expressive e.g., $\mathcal{ALC}$, DL-Lite, and RDF KBs; (iii) investigation of secrecy-preserving query answering in settings with multiple querying agents, under various restrictions on communication among agents.

References

1. Jie Bao, Giora Slutzki, and Vasant Honavar. Privacy-preserving reasoning on the semantic web. In *Web Intelligence*, pages 791–797. IEEE Computer Society, 2007.
2. Franz Baader. Terminological cycles in a description logic with existential restrictions. In *IJCAI'03: Proceedings of the 18th international joint conference on Artificial intelligence*, pages 325–330, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.
3. Sebastian Brandt. Polynomial time reasoning in a description logic with existential restrictions, gci axioms, and - what else? In Ramon López de Mántaras and Lorenza Saitta, editors, *ECAI*, pages 298–302. IOS Press, 2004.
4. Adila Krisnadhi and Carsten Lutz. Data complexity in the el family of dls. In Diego Calvanese, Enrico Franconi, Volker Haarslev, Domenico Lembo, Boris Motik, Anni-Yasmin Turhan, and Sergio Tessaris, editors, *Description Logics*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
5. K. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *J. of the Amer. Med. Informatics Assoc.*, 2000.
6. A. Rector and I. Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proceedings of the Workshop on Ontological Engineering, AAAI Spring Symposium (AAAI97), Stanford, CA*, pages 321–325, 1997.
7. Jia Tao, Giora Slutzki, and Vasant Honavar. Secrecy-preserving query answering in el. Technical Report TR10-03a, Iowa State University, Ames, IA, 2010.
8. Csilla Farkas, Alexander Brodsky, and Sushil Jajodia. Unauthorized inferences in semistructured databases. *Information Sciences*, 176:3269–3299, 2006.