Keeping Secrets in Modalized DL Knowledge Bases

Gopalakrishnan Krishnasamy Sivaprakasam[1] and Giora Slutzki[2]
[1]*Department of Mathematics & Computer Science, Central State University, Wilberforce, Ohio, U.S.A*
[2]*Department of Computer Science, Iowa State University, Ames, Iowa, U.S.A*

Keywords: Secrecy Preserving Reasoning, Knowledge Bases, Modalized Description Logic, Query Answering.

Abstract: In this paper we study Secrecy-Preserving Query Answering problem under the Open World Assumption (OWA) for $\mathcal{ELH}^{\Diamond}_{-\top}$ Knowledge Bases. Here $\mathcal{ELH}^{\Diamond}_{-\top}$ is a top-free description logic $\mathcal{ELH}$ augmented with a modal operator $\Diamond$. We employ a tableau procedure designed to compute a rooted labeled tree $\mathbb{T}$ which contains information about some assertional consequences of the given knowledge base. Given a secrecy set $\mathbb{S}$, which is a finite set of assertions, we compute a function E, called an envelope of $\mathbb{S}$, which assigns a set of assertions to each node of $\mathbb{T}$. E provides logical protection to the secrecy set $\mathbb{S}$ against the reasoning of a querying agent. Once the tree $\mathbb{T}$ and an envelope E are computed, we define the secrecy-preserving tree $\mathbb{T}_E$. Based on the information available in $\mathbb{T}_E$, assertional queries with modal operator $\Diamond$ can be answered efficiently while preserving secrecy. To the best of our knowledge, this work is first one studying secrecy-preserving reasoning in description logic augmented with modal operator $\Diamond$. When the querying agent asks a query q, the reasoner answers "Yes" if information about q is available in $\mathbb{T}_E$; otherwise, the reasoner answers "Unknown". Being able to answer "Unknown" plays a key role in protecting secrecy under OWA. Since we are not computing all the consequences of the knowledge base, answers to the queries based on just secrecy-preserving tree $\mathbb{T}_E$ could be erroneous. To fix this problem, we further augment our algorithms by providing recursive query decomposition algorithm to make the query answering procedure foolproof.

1 INTRODUCTION

Recently, Tao et al., in (Tao et al., 2014) have developed a conceptual framework to study secrecy-preserving reasoning and query answering in Description Logic (DL) Knowledge Bases (KBs) under Open World Assumptions (OWA). The approach uses the notion of an *envelope* to hide secret information against logical inference and it was first defined and used in (Tao et al., 2010). The idea behind the envelope concept is that no expression in the envelope can be logically deduced from information outside the envelope. This approach is based on the assumption that the information contained in a KB is incomplete (by OWA) and (so far) it has been restricted to very simple DLs and simple query languages. Specifically, in (Tao et al., 2010; Tao et al., 2014; Krishnasamy Sivaprakasam and Slutzki, 2016) the main idea was to utilize the secret information within the reasoning process, but then answering "Unknown" whenever the answer is truly unknown or in case the true answer could compromise confidentiality. In this paper, we extend this approach to a DL that incorporates a modal operator.

Generally, modalized DLs are DLs with modal operators. Lutz et al., in (Lutz et al., 2001) presented an exponential time tableau decision algorithm for modalized $\mathcal{ALC}$. In (Tao et al., 2012), the authors presented a PSPACE algorithm for satisfiability reasoning problem in acyclic modalized $\mathcal{ALC}$ KBs. Modal logic was used to study privacy related reasoning tasks, see (Barth and Mitchell, 2005; Halpern and O'Neill, 2005; Jafari et al., 2011). Specifically in (Halpern and O'Neill, 2005), the authors showed that the modal logic of knowledge for multiagent systems provides a framework for reasoning about anonymity. This framework was extended in (Tsukada et al., 2009) to reasoning about privacy. In an attempt to reduce the complexity of reasoning in modal logic to polynomial time, Hemaspaandra in (Hemaspaandra, 2000) had considered several propositional modal logic languages with one modal operator. Motivated by these works, in this paper we study secrecy-preserving query answering problem for $\mathcal{ELH}^{\Diamond}_{-\top}$ KBs where $\mathcal{ELH}^{\Diamond}_{-\top}$ is the top-free description logic $\mathcal{ELH}$ augmented with the modal operator $\Diamond$. The reason for excluding $\top$ from the syn-

Krishnasamy Sivaprakasam G. and Slutzki G.
Keeping Secrets in Modalized DL Knowledge Bases.
DOI: 10.5220/0006202505910598
In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017)*, pages 591-598
ISBN: 978-989-758-220-2

tax of $\mathcal{ELH}$ logic is to avoid computing tautological statements that are not relevant to secrecy preservation. In the literature there are several top-free DL languages. For instance, *DL-Lite*$_{\mathcal{R}}$ is a top-free DL, see (Calvanese et al., 2007). The syntax and semantics of the $\mathcal{ELH}^{\Diamond}_{-\top}$ DL are presented in Section 2.

Given an $\mathcal{ELH}^{\Diamond}_{-\top}$ KB $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, as a first step in constructing secrecy-preserving reasoning system, we use a tableau algorithm to compute a finite rooted labeled tree $\mathbb{T}$. The labeling set of the root node of the $\mathbb{T}$ is $\mathcal{A}^*$ which contains a set of consequences of the KB Σ, restricted to concepts that actually occur in Σ and an extra "auxiliary" set of concepts defined over the signature of Σ. Since the computed tree does not contain all the consequences of the KB, in order to answer user queries we have designed a recursive algorithm which breaks the queries into smaller assertions all the way until the information in $\mathbb{T}$ can be used.

To protect the secret information in the secrecy set $\mathbb{S}$, we extend the idea of *envelope* (as a set of assertions) to a function E that assigns a set of assertions to each node in $\mathbb{T}$. This envelope is computed by another tableau algorithm based on the idea of inverting the local and global expansion rules given in the first tableau algorithm. Once such envelope is computed, the answers to the queries are censored dependent upon the labeling set assigned by E to the nodes of $\mathbb{T}$. Since, generally, an envelope is not unique, the developer has some freedom to output a envelope (from the available choices) satisfying the needs of application domain, company policy, social obligations or user preferences.

Next, we discuss a query answering procedure which allows us to answer queries without revealing secrets. The queries are answered based on the information available in the secrecy-preserving tree obtained from the tree $\mathbb{T}$ and the envelope E, see Section 4. This tree, once computed, remains fixed. Usually in secrecy-preserving query answering framework queries are answered by checking their membership in a previously computed set, see (Tao et al., 2010; Tao et al., 2014; Krishnasamy Sivaprakasam and Slutzki, 2016). Since the secrecy-preserving tree does not contain all the statements entailed by Σ, we need to extend the query answering procedure from just membership checking. Towards that end we have designed a recursive algorithm to answer more complicated queries. To answer a query q, the algorithm first checks if q is a member of the labeling set of the root node of the secrecy-preserving tree, in which case the answer is "Yes"; otherwise, the given query is broken into subqueries based on the logical constructors, and the algorithm is applied recursively on the subqueries, see Section 5.

2 SYNTAX AND SEMANTICS

A vocabulary of $\mathcal{ELH}^{\Diamond}_{-\top}$ is a triple $< N_O, N_C, N_R >$ of countably infinite, pairwise disjoint sets. The elements of N_O are called *object* (or *individual) names*, the elements of N_C are called *concept names* and the elements of N_R are called *role names*. The set of $\mathcal{ELH}^{\Diamond}_{-\top}$ *concepts* is denoted by $\mathcal{C}$ and is defined by the following rules

$$C ::= A \mid C \sqcap D \mid \exists r.C \mid \Diamond C$$

where $A \in N_C$, $r \in N_R$, $C, D \in \mathcal{C}$ and $\Diamond C$ denotes the *modal constructor*, read as *"diamond C"*. *Assertions* are expressions of the form $C(a)$ or $r(a,b)$, *general concept inclusions (GCIs)* are expressions of the form $C \sqsubseteq D$ and *role inclusions* are expressions of the form $r \sqsubseteq s$ where $C, D \in \mathcal{C}$, $r, s \in N_R$ and $a, b \in N_O$.

The semantics of $\mathcal{ELH}^{\Diamond}_{-\top}$ concepts is defined by using Kripke structures (Kripke, 1963). A *Kripke structure* is a tuple $\mathbb{M} = \langle S, \pi, \mathcal{E} \rangle$ where S is a set of *states*, $\mathcal{E} \subseteq S \times S$ is the *accessibility* relation, and π interprets the syntax of $\mathcal{ELH}^{\Diamond}_{-\top}$ at each state $s \in S$. Further, we denote by $\mathcal{E}(s)$ the set $\{t \mid (s,t) \in \mathcal{E}\}$ of the successors of the state s. All the concepts and role names will be interpreted in a *common non-empty domain* which we denote by Δ, see (Lutz et al., 2001; Tao et al., 2012). The interpretation of concepts and role names is defined inductively as follows: for all $a \in N_O$, $A \in N_C$, $r \in N_R$, $C, D \in \mathcal{C}$ and for all $s \in S$,

$$a^{\pi(s)} \in \Delta; \quad A^{\pi(s)} \subseteq \Delta; \quad r^{\pi(s)} \subseteq \Delta \times \Delta;$$
$$(C \sqcap D)^{\pi(s)} = C^{\pi(s)} \cap D^{\pi(s)}; \ (\Diamond C)^{\pi(s)} = \bigcup_{t \in \mathcal{E}(s)} C^{\pi(t)};$$
$$(\exists r.C)^{\pi(s)} = \{d \in \Delta \mid \exists e \in C^{\pi(s)} : (d,e) \in r^{\pi(s)}\}.$$

An *ABox* $\mathcal{A}$ is a finite, non-empty set of assertions, a *TBox* $\mathcal{T}$ is a finite set of GCIs and an *RBox* $\mathcal{R}$ is a finite set of role inclusions. An $\mathcal{ELH}^{\Diamond}_{-\top}$ KB is a triple $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ where $\mathcal{A}$ is an ABox, $\mathcal{T}$ is a TBox and $\mathcal{R}$ is an RBox.

Let $\mathbb{M} = \langle S, \pi, \mathcal{E} \rangle$ be a Kripke structure, $s \in S$, $C, D \in \mathcal{C}$, $r, t \in N_R$ and $a, b \in N_O$. We say that $(\mathbb{M}, s)$ *satisfies* $C(a)$, $r(a,b)$, $C \sqsubseteq D$ or $r \sqsubseteq t$, notation $(\mathbb{M}, s) \models C(a)$, $(\mathbb{M}, s) \models r(a,b)$, $(\mathbb{M}, s) \models C \sqsubseteq D$ or $(\mathbb{M}, s) \models r \sqsubseteq t$ if, respectively, $a^{\pi(s)} \in C^{\pi(s)}$, $(a^{\pi(s)}, b^{\pi(s)}) \in r^{\pi(s)}$, $C^{\pi(s)} \subseteq D^{\pi(s)}$ or $r^{\pi(s)} \subseteq t^{\pi(s)}$. $(\mathbb{M}, s)$ *satisfies* Σ, notation $(\mathbb{M}, s) \models \Sigma$, if $(\mathbb{M}, s)$ satisfies all the assertions in $\mathcal{A}$, all the GCIs in $\mathcal{T}$ and all the role inclusions in $\mathcal{R}$. $\mathbb{M}$ *satisfies* Σ, or $\mathbb{M}$ is a *model* of Σ, if there exists a $s \in S$ such that $(\mathbb{M}, s) \models \Sigma$ and for all $t \in S$, $(\mathbb{M}, t) \models \mathcal{T} \cup \mathcal{R}$. Let α be an assertion, a

GCI or a role inclusion. We say that Σ *entails* α, notation $\Sigma \models \alpha$, if for all Kripke structures $\mathbb{M}$ satisfying Σ and for all states s of $\mathbb{M}$, $(\mathbb{M},s) \models \Sigma \Rightarrow (\mathbb{M},s) \models \alpha$.

3 COMPUTATION OF A MODEL FOR $\mathcal{ELH}^{\Diamond}_{-\top}$ KB Σ AND $\mathcal{A}^*$

Denote by N_Σ the set of all concept names and role names occurring in Σ and let $\mathcal{S}$ be a finite set of concepts over N_Σ[1]. Let $\mathcal{C}_{\Sigma,\mathcal{S}}$ be the set of all subconcepts of concepts that occur in $\mathcal{S}$ or Σ and define

$$\mathcal{A}^* = \{C(a) \mid C \in \mathcal{C}_{\Sigma,\mathcal{S}} \text{ and } \Sigma \models C(a)\} \cup \{r(a,b) \mid \Sigma \models r(a,b)\}.$$

We use $\mathcal{O}_\Sigma$ to denote the set of individual names that occur in Σ, and define the *witness set* $\mathcal{W} = \{w_C^r \mid r \in N_R \cap N_\Sigma \text{ and } C \in \mathcal{C}_{\Sigma,\mathcal{S}}\}$. Define $\mathcal{O}^* = \mathcal{O}_\Sigma \cup \mathcal{W}$. Given Σ and $\mathcal{C}_{\Sigma,\mathcal{S}}$, we outline a procedure that computes a tree called a *constraint tree over* Σ: a rooted tree $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L}\rangle$ where $\mathbb{V}$ is a set of nodes, $k_0 \in \mathbb{V}$ is the root of $\mathbb{T}$, $\mathbb{E}$ is a set of directed edges and $\mathbb{L}$ is a function that labels each node with a set of assertions obtained by applying the *expansion rules* specified below. The procedure builds $\mathbb{T}$ starting from the root node k_0 whose labeling set $\mathbb{L}(k_0)$ is initialized as the ABox $\mathcal{A}$. Further, $\mathbb{T}$ is grown by recursively applying the expansion rules in Figures 1 and 2. $\mathbb{T}$ is said to be *completed* if no expansion rule in Figures 1 or 2 is applicable to it. The procedure is designed to output a completed constraint tree $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L}\rangle$ with $\mathbb{L}(k_0) = \mathcal{A}^*$. For the purpose of query answering, $\mathbb{T}$ is used as a "good approximation" of a (Kripke) model of the given KB, for details see Section 5.

In more detail, there are two kinds of expansion rules: (a) *local* expansion rules and (b) *global* expansion rules. Local expansion rules are given in Figure 1 and generate new assertions within a single labeling set. The $\sqcap^-$-rule decomposes conjunctions, and $\exists^-$-rule decomposes existential restriction assertions of the form $\exists r.C(a)$ by introducing a corresponding witness w_C^r from the set $\mathcal{W}$. The $\sqsubseteq$-rule derives new assertions based on the GCIs present in $\mathcal{T}$. To construct concept assertions whose associated concept expressions already belong to $\mathcal{C}_{\Sigma,\mathcal{S}}$, we use the $\sqcap^+$ and $\exists^+$-rules. Finally, the H-rule derives role assertions based on role inclusions in $\mathcal{R}$. The global expansion rules are given in Figure 2. The $\Diamond^-$-rule generates new nodes that are directly accessible from the current node. The $\Diamond^+$-rule adds a new $\Diamond$ assertion to the parent node of the current node.

[1]A technicality; $\mathcal{S}$ will be used in Section 4 in the context of secrecy-preserving reasoning.

$\sqcap^+ -$ rule : if $C(a)$, $D(a) \in \mathbb{L}(i)$, $C \sqcap D \in \mathcal{C}_{\Sigma,\mathcal{S}}$, and $C \sqcap D(a) \notin \mathbb{L}(i)$, then $\mathbb{L}(i) := \mathbb{L}(i) \cup \{C \sqcap D(a)\}$;

$\sqcap^- -$ rule : if $C \sqcap D(a) \in \mathbb{L}(i)$, and $C(a) \notin \mathbb{L}(i)$ or $D(a) \notin \mathbb{L}(i)$, then $\mathbb{L}(i) := \mathbb{L}(i) \cup \{C(a), D(a)\}$;

$\exists^+ -$ rule : if $r(a,b)$, $C(b) \in \mathbb{L}(i)$, $\exists r.C \in \mathcal{C}_{\Sigma,\mathcal{S}}$ and $\exists r.C(a) \notin \mathbb{L}(i)$, then $\mathbb{L}(i) := \mathbb{L}(i) \cup \{\exists r.C(a)\}$;

$\exists^- -$ rule : if $\exists r.C(a) \in \mathbb{L}(i)$, and $\forall b \in \mathcal{O}^*, \{r(a,b), C(b)\} \not\subseteq \mathbb{L}(i)$, then $\mathbb{L}(i) := \mathbb{L}(i) \cup \{r(a, w_C^r), C(w_C^r)\}$, where $w_C^r \in \mathcal{W}$;

$\sqsubseteq -$ rule : if $C(a) \in \mathbb{L}(i)$, $C \sqsubseteq D \in \mathcal{T}$, and $D(a) \notin \mathbb{L}(i)$, then $\mathbb{L}(i) := \mathbb{L}(i) \cup \{D(a)\}$;

$H -$ rule : if $r(a,b) \in \mathbb{L}(i)$, $r \sqsubseteq s \in \mathcal{R}$, and $s(a,b) \notin \mathbb{L}(i)$, then $\mathbb{L}(i) := \mathbb{L}(i) \cup \{s(a,b)\}$;

Figure 1: Local expansion rules.

$\Diamond^- -$ rule : if there is a node i with $\Diamond C(a) \in \mathbb{L}(i)$ and i has no successor j with $C(a) \in \mathbb{L}(j)$, then add a new successor k of i with $\mathbb{L}(k) := \{C(a)\}$;

$\Diamond^+ -$ rule : if there is a node i with $C(a) \in \mathbb{L}(i)$, $\Diamond C \in \mathcal{C}_{\Sigma,\mathcal{S}}$ and i has a parent j with $\Diamond C(a) \notin \mathbb{L}(j)$, then $\mathbb{L}(j) := \mathbb{L}(j) \cup \{\Diamond C(a)\}$.

Figure 2: Global expansion rules.

Example 1. *Let* $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R}\rangle$ *be a* $\mathcal{ELH}^{\Diamond}_{-\top}$ *KB, where* $\mathcal{A} = \{\Diamond A(a), C(d), u(a,d)\}$, $\mathcal{T} = \{\Diamond A \sqsubseteq B, C \sqsubseteq \Diamond\Diamond(D \sqcap E), E \sqsubseteq \exists u.F, \Diamond D \sqsubseteq \Diamond G\}$, $\mathcal{R} = \{u \sqsubseteq v\}$ *and* $\mathcal{S} = \{\exists u.C, \Diamond\Diamond D\}$. *Then, applying the rules in Figures 1 and 2 we compute the completed constraint tree* $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L}\rangle$ *whose labeling sets are given in Figure 3. In this example,* $\mathcal{O}^* = \{a, d, w_C^u, w_F^u\}$ *and*

- $\mathbb{L}(k_0) = \{\Diamond A(a), B(a), C(d), u(a,d), \exists u.C(a), v(a,d), \Diamond\Diamond(D \sqcap E)(d), \Diamond\Diamond D(d)\}$,
- $\mathbb{L}(k_1) = \{A(a)\}$,
- $\mathbb{L}(k_2) = \{\Diamond(D \sqcap E)(d), \Diamond D(d), \Diamond G(d)\}$,
- $\mathbb{L}(k_3) = \{D \sqcap E(d), D(d), E(d), \exists u.F(d), u(d, w_F^u), F(w_F^u), v(d, w_F^u)\}$ *and*
- $\mathbb{L}(k_4) = \{G(d)\}$. □

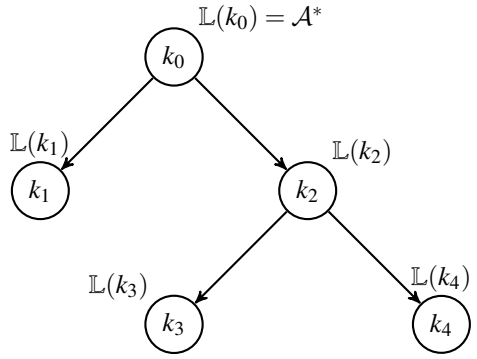


Figure 3: Completed constraint tree $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L} \rangle$.

We will use the following notion of TBox acyclicity, called here $\Diamond$-*acyclicity*.

Definition 1. *A sequence $S_0, S_1 \ldots, S_n, \ldots$ of concept assertions over Σ, is called a $\Diamond$-sequence, if it satisfies the following conditions:*

- *$S_0 = C_0(a_0)$, $C_0 \in \mathcal{C}_{\Sigma,\mathbb{S}}$, $a_0 \in \mathcal{O}^*$.*
- *Given, $S_n = C_n(a_n)$, with $C_n \in \mathcal{C}_{\Sigma,\mathbb{S}}$, $a_n \in \mathcal{O}^*$, the next element in the sequence can be obtained as follows: Let $\mathcal{B}_n$ be the set of all assertions obtained by applying the local rules starting from S_n. Put $\mathcal{D}_n = \mathcal{B}_n \cup \{S_n\}$.*
  - *If $\mathcal{D}_n$ does not contain any $\Diamond$-assertions, then S_n is the last assertion of the sequence.*
  - *If $\mathcal{D}_n$ contains some $\Diamond$-assertions, then pick one, say $\Diamond P(b)$, and define $S_{n+1} = C_{n+1}(a_{n+1}) = P(b)$.*

The resulting $\Diamond$-sequence is said to be non-repetitive, if for distinct i, j, $C_i \neq C_j$.

Definition 2. *A TBox $\mathcal{T}$ is said to be $\Diamond$-acyclic (with respect to the rules given in Figures 1 and 2), if every $\Diamond$-sequence is non-repetitive.*

In this paper, we assume that all TBoxes are $\Diamond$-acyclic as per Definition 2 (we shall omit the phrase "with respect to the rules"). We denote by Λ the *algorithm* which, given Σ and $\mathcal{C}_{\Sigma,\mathcal{S}}$, nondeterministically applies the expansion rules in Figures 1 and 2 until no further applications are possible. It is easy to see that for each node k in the constraint tree $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L} \rangle$, the size of $\mathbb{L}(k)$ is polynomial in $|\Sigma| + |\mathcal{C}_{\Sigma,\mathcal{S}}|$. An upper bound for the depth of $\mathbb{T}$ is given in the following claim which follows immediately from Definitions 1 and 2.

Claim 1. *The depth of $\mathbb{T}$ is $O(|\mathcal{C}_{\Sigma,\mathcal{S}}|)$.*

All executions of Λ terminate and by Claim 1, Λ builds a tree $\mathbb{T}$ whose depth is linear in $|\mathcal{C}_{\Sigma,\mathbb{S}}|$. Since the $\Diamond$-rule can in some cases be applied exponentially many times in $|\Sigma| + |\mathcal{C}_{\Sigma,\mathbb{S}}|$, $\mathbb{T}$ may have exponentially many nodes. For instance, consider a $\mathcal{ELH}^{\Diamond}_{-\top}$ KB $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, where $\mathcal{A} = \{A_1(a)\}$, $\mathcal{T} = \{A_i \sqsubseteq \Diamond \exists r.A_{i+1}, A_i \sqsubseteq \Diamond \exists s.A_{i+1}, 1 \leq i \leq n-1\}$ and $\mathcal{R} = \emptyset$. Clearly, TBox $\mathcal{T}$ is $\Diamond$-acyclic. To compute the constraint tree $\mathbb{T}$ for Σ, the global rules must be applied exponentially many times, implying that, the worst case the running time of Λ is exponential in $|\Sigma| + |\mathcal{C}_{\Sigma,\mathbb{S}}|$.

Before proving the correctness of Λ, we define the notion of interpretation of a constraint tree, see (Lutz et al., 2001; Tao et al., 2012).

Definition 3. *Let $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L} \rangle$ be a constraint tree over Σ, $\mathbb{M} = \langle S, \pi, \mathcal{E} \rangle$ a Kripke structure, and σ a mapping from $\mathbb{V}$ to S. We say that $\mathbb{M}$ satisfies $\mathbb{T}$ via σ if for all $k, k' \in \mathbb{V}$,*

- *$(k, k') \in \mathbb{E} \Rightarrow (\sigma(k), \sigma(k')) \in \mathcal{E}$, and*
- *$(\mathbb{M}, \sigma(k)) \models \mathbb{L}(k)$, i.e., $(\mathbb{M}, \sigma(k)) \models \alpha$ for every $\alpha \in \mathbb{L}(k)$*

We say that $\mathbb{M}$ satisfies $\mathbb{T}$, denoted as $\mathbb{M} \,|\!|\!=\, \mathbb{T}$, if there is a mapping σ such that $\mathbb{M}$ satisfies $\mathbb{T}$ via σ.

In the next lemma, we formulate the local soundness property of Λ. We say that f' is an extension of a function f if f' agrees with f on the domain of f. The proof of the following lemma is omitted.

Lemma 1. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be $\mathcal{ELH}^{\Diamond}_{-\top}$ KB with an $\Diamond$-acyclic TBox $\mathcal{T}$ and let $\mathbb{M} = \langle S, \pi, \mathcal{E} \rangle$ be a Kripke structure satisfying Σ. Also let $\mathbb{T}$ be a constraint tree over Σ, α a local or global expansion rule and $\mathbb{T}_\alpha$ a constraint tree obtained by applying α to $\mathbb{T}$. If $\mathbb{M}$ satisfies $\mathbb{T}$ via σ, then there exists a Kripke structure $\mathbb{M}' = \langle S, \pi', \mathcal{E} \rangle$ such that*

- *$\mathbb{M}'$ satisfies Σ and π' is an extension of π,*
- *$\mathbb{M}'$ satisfies $\mathbb{T}_\alpha$ via σ', an extension of σ.*

Lemma 1 makes sure that each application of local and global rules preserves the model existence property. Next we define the *canonical Kripke structure* of a constraint tree.

Definition 4. *Let $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L} \rangle$ be a completed constraint tree over Σ. The canonical Kripke structure $\mathbb{M}_\mathbb{T} = \langle S, \pi, \mathcal{E} \rangle$ for $\mathbb{T}$ is defined as follows:*

- *$S = \mathbb{V}$, $\mathcal{E} = \mathbb{E}$, $\Delta = \mathcal{O}^* = \mathcal{O}_\Sigma \cup \mathcal{W}$,*
- *$a^{\pi(k)} = a$ for all $a \in \mathcal{O}^*$ and each $k \in \mathbb{V}$,*
- *$A^{\pi(k)} = \{a \in \mathcal{O}^* \mid A(a) \in \mathbb{L}(k)\}$, $A \in N_C \cap N_\Sigma$,*
- *$r^{\pi(k)} = \{(a,b) \in \mathcal{O}^* \times \mathcal{O}^* \mid r(a,b) \in \mathbb{L}(k)\}$, for all $r \in N_R \cap N_\Sigma$,*

$\pi(k)$ is extended to compound concepts in the usual way (see Section 2).

The following lemma shows that $\mathbb{M}_\mathbb{T}$ satisfies the completed constraint tree $\mathbb{T}$. The proof is omitted.

Lemma 2. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be $\mathcal{ELH}^{\Diamond}_{-\top}$ KB with a $\Diamond$-acyclic TBox $\mathcal{T}$. Also let $\mathbb{T}$ be a completed constraint tree over Σ. Then $\mathbb{M}_\mathbb{T} \,|\!|\!=\, \mathbb{T}$.*

Next we prove that $(\mathbb{M}_\mathbb{T}, k) \models \mathcal{T} \cup \mathcal{R}$, for each $k \in S$. We need the following auxiliary lemma whose proof is an easy induction on the structure of C.

Lemma 3. *For each $C \in \mathcal{C}_{\Sigma,\mathcal{S}}$, each $a \in \mathcal{O}^*$ and each $k \in \mathbb{V}$, if $(\mathbb{M}_\mathbb{T}, k) \models C(a)$ then $C(a) \in \mathbb{L}(k)$.*

The proof of the following lemma follows immediately from Lemmas 2 and 3 and it is omitted.

Lemma 4. *For each $k \in S$, $(\mathbb{M}_\mathbb{T}, k) \models \mathcal{T} \cup \mathcal{R}$.*

The following corollary is an immediate consequence of Lemmas 2 and 4.

Corollary 1. *$\mathbb{M}_\mathbb{T}$ satisfies Σ.*

Proof. By Definitions 3 and 4 and Lemmas 2 and 4, we have that (1) $(\mathbb{M}_\mathbb{T}, k_0) \models \Sigma$ and (2) for each $k \in \mathbb{V}$, $(\mathbb{M}_\mathbb{T}, k) \models \mathcal{T} \cup \mathcal{R}$. Hence $\mathbb{M}_\mathbb{T}$ satisfies Σ. □

The proof of the next theorem follows from Definition 4 and Lemma 3. In a sense, this theorem captures the completeness property of the algorithm Λ.

Theorem 1. *Let $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L} \rangle$ be a completed constraint tree over Σ and $\mathbb{M}_\mathbb{T} = \langle S, \pi, \mathcal{E} \rangle$ a canonical Kripke structure for $\mathbb{T}$. Then, for all $k \in \mathbb{V}$, $C \in \mathcal{C}_{\Sigma,\mathcal{S}}$, $r \in N_\Sigma \cap N_R$, and all $a, b \in \mathcal{O}^*$*

- *$(\mathbb{M}_\mathbb{T}, k) \models r(a,b) \Rightarrow r(a,b) \in \mathbb{L}(k)$ and*
- *$(\mathbb{M}_\mathbb{T}, k) \models C(a) \Rightarrow C(a) \in \mathbb{L}(k)$.*

Finally, the following is a consequence of Theorem 1 and Corollary 1.

Corollary 2. *$\mathbb{L}(k_0) = \mathcal{A}^*$.*

4 SECRECY-PRESERVING REASONING IN $\mathcal{ELH}^{\Diamond}_{-\top}$ KB'S

Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be a $\mathcal{ELH}^{\Diamond}_{-\top}$ KB and $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L} \rangle$ a completed constraint tree over Σ. Let $\mathbb{S} \subseteq \mathcal{A}^*$ be the "secrecy set". Given Σ, $\mathbb{S}$ and $\mathbb{T}$, the objective is to answer assertion queries while preserving secrecy, i.e., answering queries so that assertions in $\mathbb{S}$ remain protected. Our approach is to compute a function E that assigns a finite set of assertions to each node in $\mathbb{T}$. E is called the *secrecy Envelope* for $\mathbb{S}$, so that protecting $E(i)$ for all $i \in \mathbb{V}$, the querying agent cannot logically infer any assertion in $\mathbb{S}$. The sets $E(i)$ for each $i \in \mathbb{V}$ are obtained by applying the *inverted expansion rules* given in Figures 4 and 5. The role of OWA in answering the queries is the following: When answering a query with "Unknown", the querying agent should not be able to distinguish between the case that the answer to the query is truly unknown to the KB reasoner and the case that the answer is being protected for reasons of secrecy. We envision a situation in which once the $\mathbb{T}$ is computed, a *reasoner* $\Re$ is associated with it, i.e., $\Re$ has unfettered access to $\mathbb{T}$. $\Re$ is designed to answer queries as follows: If a query cannot be inferred from Σ, the answer is "Unknown". If it can be inferred and it is not in $E(k_0)$, the answer is "Yes"; otherwise, the answer is "Unknown". We make the following assumptions about the capabilities of the querying agent:

(a) does not have direct access to ABox $\mathcal{A}$, but is aware of the underlying vocabulary of Σ,
(b) has full access to TBox $\mathcal{T}$ and RBox $\mathcal{R}$,
(c) can ask queries in the form of assertions, and
(d) is not aware of the witness set $\mathcal{W}$, by *hidden name assumptions* (HNA), for more details see (Tao et al., 2010).

We formally define the notion of an envelope

Definition 5. *Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be a $\mathcal{ELH}^{\Diamond}_{-\top}$ KB, $\mathbb{S}$ a finite secrecy set and $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L} \rangle$ a completed constraint tree. The secrecy envelope of $\mathbb{S}$ is a function E with domain $\mathbb{V}$ satisfying the following properties:*

- *$\mathbb{S} \subseteq E(k_0)$; for each $i \in \mathbb{V}$, $E(i) \subseteq \mathbb{L}(i)$, and*
- *for each $i \in \mathbb{V}$, each $\alpha \in E(i)$, $\mathbb{L}(i) \setminus E(i) \not\models \alpha$.*

The intuition for the above definition is that for each $i \in \mathbb{V}$, no information in $E(i)$ can be inferred from the set $\mathbb{L}(i) \setminus E(i)$. To compute an envelope, we use the idea of inverting the rules of Figures 1 and 2 (see (Tao et al., 2010), where this approach was first utilized for membership assertions). Induced by the Local and Global expansion rules in Figures 1 and 2, we define the corresponding "inverted" Local and Global expansion rules in Figures 4 and 5, respectively. Note that the $\exists^-$-rule does not have its corresponding inverted rule. The reason for the omission is that an application of this rule results in adding assertions with individual names from the witness set. By HNA, the querying agent is barred from asking any queries that involve individual names from the witness set. Inverted expansion rules are denoted by prefixing Inv- to the name of the corresponding expansion rules.

From now on, we assume that $\mathbb{T}$ has been computed and is readily available for computing the envelope. The computation begins with the initialization step: The set $E(k_0)$ is initialized as $\mathbb{S}$, and $E(i)$ is initialized as $\emptyset$ for all $i \in \mathbb{V} \setminus \{k_0\}$. Next, the sets $E(k_0)$ and $E(i)$ for all $i \in \mathbb{V} \setminus \{k_0\}$ are expanded using the inverted expansion rules listed in Figures 4 and 5 until no further applications are possible. The resulting function E is said to be *completed*. We denote by $\Lambda_\mathbb{S}$ the algorithm which computes the sets $E(i)$ for all $i \in \mathbb{V}$. Due to non-determinism in applying the rules Inv-$\sqcap^+$ and Inv-$\exists^+$, different executions of $\Lambda_\mathbb{S}$ may result different outputs. Since for each $i \in \mathbb{V}$, $\mathbb{L}(i)$

Inv-$\sqcap^-$ −rule : if $\{C(a),\ D(a)\} \cap E(i) \neq \emptyset$
and $C \sqcap D(a) \in \mathbb{L}(i) \setminus E(i)$,
then $E(i) := E(i) \cup \{C \sqcap D(a)\}$;

Inv-$\sqcap^+$ −rule : if $C \sqcap D(a) \in E(i)$,
$\{C(a),\ D(a)\} \subseteq \mathbb{L}(i) \setminus E(i)$
and $C \sqcap D \in \mathcal{C}_{\Sigma,\mathcal{S}}$,
then $E(i) := E(i) \cup \{C(a)\}$
or $E(i) := E(i) \cup \{D(a)\}$;

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

Inv-$\sqsubseteq$ −rule : if $D(a) \in E(i)$, $C \sqsubseteq D \in \mathcal{T}$,
and $C(a) \in \mathbb{L}(i) \setminus E(i)$,
then $E(i) := E(i) \cup \{C(a)\}$;

Inv-H − rule : if $s(a,b) \in E(i)$, $r \sqsubseteq s \in \mathcal{R}$,
and $r(a,b) \in \mathbb{L}(i) \setminus E(i)$,
then $E(i) := E(i) \cup \{r(a,b)\}$.

Figure 4: Inverted local expansion rules.

is finite, the computation of $\Lambda_{\mathbb{S}}$ terminates. Let the sets $E(i)$ for $i \in \mathbb{V}$ be an output of $\Lambda_{\mathbb{S}}$. Since the size of each $\mathbb{L}(i)$ is polynomial in $|\Sigma| + |\mathcal{C}_{\Sigma,\mathcal{S}}|$, and each application of inverted expansion rule moves an assertion from $\mathbb{L}(i)$ into $E(i)$, the size of $E(i)$ is at most the size of $\mathbb{L}(i)$. Since the size of $\mathbb{V}$ can be exponential, $\Lambda_{\mathbb{S}}$ may take exponential time to compute the sets $E(i)$. Define the *secrecy-preserving tree* (constraint) for the secrecy set $\mathbb{S}$ to be $\mathbb{T}_E = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L}_E \rangle$, where $\mathbb{L}_E(i) = \mathbb{L}(i) \setminus E(i)$ for all $i \in \mathbb{V}$.

Example 2. *(Example 1 cont.) Recall that $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L} \rangle$ is the completed constraint tree. Let $\mathbb{S} = \{B(a), \Diamond\Diamond D(d)\}$ be the secrecy set. Then, by using rules in Figures 4 and 5 we compute the envelope for $\mathbb{S}$, and one of the corresponding secrecy-preserving trees is given below:*

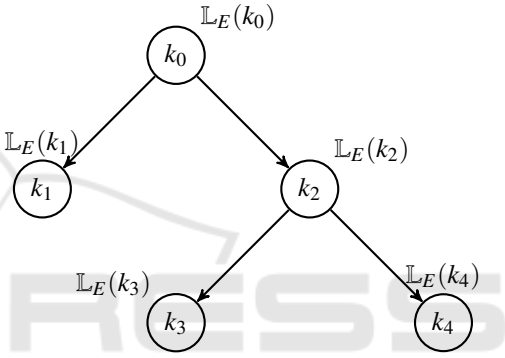
- $E(k_0) = \mathbb{S} \cup \{\Diamond A(a), C(d), \Diamond\Diamond(D \sqcap E)(d)\}$,
- $E(k_1) = \{A(a)\}$,
- $E(k_2) = \{\Diamond(D \sqcap E)(d), \Diamond D(d)\}$,
- $E(k_3) = \{D \sqcap E(d), D(d)\}$ and
- $E(k_4) = \emptyset$.

Before proving the main result on envelopes, we present several auxiliary lemmas. First shows that for each $i \in \mathbb{V}$, no assertion in $E(i)$ is "logically reachable" from $\mathbb{L}_E(i)$. The proof is omitted.

Inv-$\Diamond^-$ − rule : if there is a node j with
$C(a) \in E(j)$ and
$\Diamond C(a) \in \mathbb{L}(i) \setminus E(i)$
where i is the parent of j,
then $E(i) := E(i) \cup \{\Diamond C(a)\}$;

Inv-$\Diamond^+$ − rule : if there is a node i with
$\Diamond C(a) \in E(i)$ and
$C(a) \in \mathbb{L}(j) \setminus E(j)$
where j is a successor of i
and $\Diamond C \in \mathcal{C}_{\Sigma,\mathcal{S}}$,
then $E(j) := E(j) \cup \{C(a)\}$.

Figure 5: Inverted global expansion rule.



- $\mathbb{L}_{\mathbb{E}}(k_0) = \{u(a,d), \exists u.C(a), v(a,d)\}$,
- $\mathbb{L}_{\mathbb{E}}(k_1) = \emptyset$,
- $\mathbb{L}_{\mathbb{E}}(k_2) = \{\Diamond G(d)\}$,
- $\mathbb{L}_{\mathbb{E}}(k_3) = \{E(d), \exists u.F(d), u(d, w_F^u), F(w_F^u), v(d, w_F^u)\}$ and
- $\mathbb{L}_{\mathbb{E}}(k_4) = \{G(d)\}$. □

Figure 6: Secrecy-preserving tree $\mathbb{T}_E = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L}_E \rangle$.

Lemma 5. *Let E be a completed function resulting from the algorithm $\Lambda_{\mathbb{S}}$. Also, let $\mathbb{T}_E = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L}_E \rangle$ be a secrecy-preserving tree. Then, for each $i \in \mathbb{V}$, $\mathbb{L}_E(i)$ is completed.*

Next we claim that the secrecy-preserving tree has properties similar to those of its completed constraint tree. The proof is similar to the proofs of Lemmas 2, 3 and 4.

Lemma 6. *Let $\mathbb{T}_E = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L}_E \rangle$ be a secrecy-preserving tree obtained from the completed constraint tree $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L} \rangle$ over Σ and the completed function E. Define the canonical Kripke structure $\mathbb{M}_{\mathbb{T}}^E = \langle S, \pi, \mathcal{E} \rangle$ for $\mathbb{T}_E$ as*

- $S = \mathbb{V}$, $\mathcal{E} = \mathbb{E}$, $\Delta = \mathcal{O}^* = \mathcal{O}_\Sigma \cup \mathcal{W}$,
- $a^{\pi(k)} = a$ *for all* $a \in \mathcal{O}^*$ *and each* $k \in \mathbb{V}$,
- $A^{\pi(k)} = \{a \in \mathcal{O}^* \mid A(a) \in \mathbb{L}_E(k)\}$, *for all* $A \in N_C$,

- $r^{\pi(k)} = \{(a,b) \in \mathcal{O}^* \times \mathcal{O}^* \mid r(a,b) \in \mathbb{L}_E(k)\}$, *for all* $r \in N_R$,

$\pi(k)$ *is extended to compound concepts in the usual way (see Section 2). Then,*

- $\mathbb{M}^E_{\mathbb{T}} \mid\mid\!= \mathbb{T}_E$,
- *For each* $C \in \mathcal{C}_{\Sigma,\mathcal{S}}$, *each* $a \in \mathcal{O}^*$ *and each* $k \in \mathbb{V}$, *if* $(\mathbb{M}^E_{\mathbb{T}},k) \models C(a)$, *then* $C(a) \in \mathbb{L}_E(k)$ *and*
- *For each* $k \in S$, $(\mathbb{M}^E_{\mathbb{T}},k) \models \mathcal{T} \cup \mathcal{R}$.

Finally, we show that a completed function E is in fact an envelope for the secrecy set $\mathbb{S}$.

Theorem 2. *Let* $\mathbb{T} = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L} \rangle$ *be a completed constraint tree over* Σ. *Also, let* $\mathbb{T}_E = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L}_E \rangle$ *be a secrecy-preserving tree for the secrecy set* $\mathbb{S}$. *Then, the completed function* E *is an envelope for* $\mathbb{S}$.

Proof. We have to show that the completed function E satisfies all three properties of Definition 5. Properties 1 and 2 are obvious. To prove property 3, suppose that for some $i \in \mathbb{V}$, some $\alpha \in E(i)$, $\mathbb{L}_E(i) \models \alpha$.

Let $\mathbb{M}^E_{\mathbb{T}} = \langle S, \pi, \mathcal{E} \rangle$ be the canonical Kripke structure for $\mathbb{T}_E$. By Lemma 6, for each $i \in \mathbb{V}$, $(\mathbb{M}^E_{\mathbb{T}}, i) \models \mathbb{L}_E(i)$. Again, by Lemma 6, $\alpha \in \mathbb{L}_E(i)$. This is a contradiction. □

5 QUERY ANSWERING

Let $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ be a $\mathcal{ELH}^{\Diamond}_{-\top}$ KB. We assume that the secrecy-preserving tree $\mathbb{T}_E = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L}_E \rangle$ has been precomputed and use $\mathbb{E}(k)$ to denote the set $\{k' \in \mathbb{V} \mid (k,k') \in \mathbb{E}\}$ of the successors of the node $k \in \mathbb{V}$. The reasoner $\Re$ answers queries based on the information in $\mathbb{T}_E$ and replies to a query q with "Yes" if $\Sigma \models q$ and $q \notin E(k_0)$; otherwise, the answer is "Unknown". Recall that, because of the syntactic restrictions of the language $\mathcal{ELH}^{\Diamond}_{-\top}$, $\Re$ does not answer "No" to any query.

Since the completed constraint tree $\mathbb{T}$ over Σ does not contain all the consequences of Σ, the completed secrecy-preserving tree $\mathbb{T}_E$ obtained from $\mathbb{T}$ does not contain all the information needed to answer queries. To address this problems we provide a procedure Eval(k,q) which works by recursively decomposing the compound queries all the way to the information available in $\mathbb{T}_E$, see Figure 7. Initial call of this procedure is at the root node k_0 of $\mathbb{T}_E$. In lines 1 and 2, the reasoner checks the membership of q in $\mathbb{L}_E(k)$ and answers "Yes" if $q \in \mathbb{L}_E(k)$. From line 3 onwards we consider cases in which query q is broken into subqueries based on the constructors defined in $\mathcal{ELH}^{\Diamond}_{-\top}$ and apply the procedure recursively. The following theorem states the correctness claim of the algorithm.

```
Eval(k, q)
 1: case q ∈ L_E(k) = L(k) \ E(k)
 2:     return "Yes"
 3: case q = C ⊓ D(a)
 4:     if Eval(k, C(a)) ="Yes" and
            Eval(k, D(a)) ="Yes" then
 5:         return "Yes"
 6:     else
 7:         return "Unknown"
 8: case q = ∃r.C(a)
 9:     if for some d ∈ O* [ r(a,d) ∈ L_E(k)
            and Eval(k, C(d)) ="Yes"] then
10:         return "Yes"
11:     else
12:         return "Unknown"
13: case q = ◇C(a)
14:     if for some l ∈ E(k)
            [Eval(l, C(a)) = "Yes" ] then
15:         return "Yes"
16:     else
17:         return "Unknown"
```

Figure 7: Query answering algorithm.

Theorem 3. *Let* $\Sigma = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ *be an* $\mathcal{ELH}^{\Diamond}_{-\top}$ *KB,* $\mathbb{T}_E = \langle \mathbb{V}, k_0, \mathbb{E}, \mathbb{L}_E \rangle$ *a completed secrecy-preserving tree and* q *a query. Then, for every* $k \in \mathbb{V}$,

- *Soundness: Eval*(k,q) *outputs "Yes"* $\Rightarrow \mathbb{L}_E(k) \models q$;
- *Completeness: Eval*(k,q) *outputs "Unknown"* $\Rightarrow \mathbb{L}_E(k) \not\models q$.

Proof. We omit the proof of soundness. To prove the completeness part assume that $\mathbb{L}_E(k) \models q$. We have to show that Eval(k,q) = "Yes". Let $\mathbb{M}^E_{\mathbb{T}}$ be the canonical Kripke structure for $\mathbb{T}_E$ as defined in Section 4. By Lamma 6, $\mathbb{M}^E_{\mathbb{T}} \mid\mid\!= \mathbb{T}_E$ and for all $k \in \mathbb{V}$, $(\mathbb{M}^E_{\mathbb{T}},k) \models \mathcal{T} \cup \mathcal{R}$. Therefore $(\mathbb{M}^E_{\mathbb{T}},k) \models \mathbb{L}_E(k)$ and hence, by the assumption, for every k, $(\mathbb{M}^E_{\mathbb{T}},k) \models q$. We prove the claim by induction on the structure of q. The inductive hypothesis is, for each $k \in \mathbb{V}$ and each assertion α if $(\mathbb{M}^E_{\mathbb{T}},k) \models \alpha$, then Eval$(k,\alpha)$ = "Yes". The base case: Let $q = C(a)$ where $C \in \mathcal{C}_{\Sigma,\mathcal{S}}$. Then, by Lemma 6, $C(a) \in \mathbb{L}_E(k)$. By Lines 1 and 2 in Figure 7, the claim follows immediately. Next, let $q = C(a)$ where $C \notin \mathcal{C}_{\Sigma,\mathcal{S}}$.

- $q = C \sqcap D(a)$. To answer this query the algorithm computes Eval$(k,C(a))$ and Eval$(k,D(a))$. Now, the assumption $(\mathbb{M}^E_{\mathbb{T}},k) \models C \sqcap D(a)$ implies $(\mathbb{M}^E_{\mathbb{T}}, k) \models C(a)$ and $(\mathbb{M}^E_{\mathbb{T}},k) \models D(a)$ which, by inductive hypothesis, implies that Eval$(k,C(a))$ = Eval$(k,D(a))$ = "Yes". Hence, by Lines 4 and 5 in Figure 7, Eval$(k,C \sqcap D(a))$ = "Yes".
- $q = \exists r.C(a)$. By the assumption, $(\mathbb{M}^E_{\mathbb{T}},k) \models \exists r.C(a)$. This implies that, for some $d \in$

$\mathcal{O}^*, (\mathbb{M}^E_{\mathbb{T}}, k) \models r(a,d)$ and $(\mathbb{M}^E_{\mathbb{T}}, k) \models C(d)$. By Theorem 1, $r(a,d) \in \mathbb{L}_E(k)$ and by the inductive hypothesis Eval$(k, C(d))$="Yes". Hence, by the Lines 9 and 10 in Figure 7, Eval$(k, \exists r.C(a))$= "Yes".

- $q = \Diamond C(a)$. Then, $(\mathbb{M}^E_{\mathbb{T}}, k) \models \Diamond C(a)$. This implies that, for some $l \in \mathcal{E}(k)$, $(\mathbb{M}^E_{\mathbb{T}}, l) \models C(a)$. By Definition 4, $(k,l) \in \mathbb{E}$ and therefore $l \in \mathbb{E}(k)$. By the inductive hypothesis Eval$(l, C(a))$ = "Yes". Hence, by the Lines 14 and 15 in Figure 7, Eval$(k, \Diamond C(a))$= "Yes".

□

Given an assertional query q, the algorithm given in Figure 7 checks for some assertions related to query q in the labeling sets of nodes along a particular path in $\mathbb{T}_E$. Since the size of each labeling set is bounded by $|\Sigma| + |\mathcal{C}_{\Sigma,\mathcal{S}}|$, by the Claim 1, this algorithm runs in time polynomial in $|\Sigma| + |\mathcal{C}_{\Sigma,\mathcal{S}}|$. Hence the assertional query answering can be done in polynomial time in the size of $|\Sigma| + |\mathcal{C}_{\Sigma,\mathcal{S}}|$.

Example 3. *(example 2 cont.) Recall that $\mathbb{T}_E$ is a secrecy-preserving tree. Suppose that the querying agent asks the assertional queries $\exists u.C(a)$, $\Diamond\Diamond\exists v.F(d)$ and $\Diamond A(a)$. Using the algorithm in Figure 7, we get the following answers:*

q	Eval(k,q)	Remarks
$\exists u.C(a)$	Yes	by Lines 1 and 2
$\Diamond\Diamond\exists v.F(d)$	Yes	by 14, 15, 9 and 1
$\Diamond A(a)$	Unknown	by 14 and 17 □

6 CONCLUSIONS

In this paper we have studied the problem of secrecy-preserving query answering over $\mathcal{ELH}^{\Diamond}_{-\top}$ KBs. We have used the conceptual logic-based framework for secrecy-preserving reasoning which was introduced in Tao et al. 2014, to a top-free description logic $\mathcal{ELH}$ augmented with a modal operator $\Diamond$. The main contribution is in the way that we compute the consequences and preserve secrecy while answering queries. We break the process into two parts, the first one using the $\Diamond$-assertions in the KB, precomputes the rooted labeled tree $\mathbb{T}$ and the envelope E for the given secrecy set $\mathbb{S}$. For this we use two separate (but related) tableau procedures. In query answering step, given $\mathbb{T}$ and E, we define the secrecy-preserving tree $\mathbb{T}_E$. Once $\mathbb{T}_E$ has been computed, the query answering procedure is efficient and can be implemented in polynomial time.

REFERENCES

Barth, A. and Mitchell, J. C. (2005). Enterprise privacy promises and enforcement. In *Proceedings of the 2005 workshop on Issues in the theory of security*, pages 58–66. ACM.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. (2007). Tractable reasoning and efficient query answering in description logics: The dl-lite family. *J. of Automated Reasoning*, 39(3):385–429.

Halpern, J. Y. and O'Neill, K. R. (2005). Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 13(3):483–514.

Hemaspaandra, E. (2000). The complexity of poor mans logic. In *STACS 2000*, pages 230–241. Springer.

Jafari, M., Fong, P. W., Safavi-Naini, R., Barker, K., and Sheppard, N. P. (2011). Towards defining semantic foundations for purpose-based privacy policies. In *Proceedings of the first ACM conference on Data and application security and privacy*, pages 213–224. ACM.

Kripke, S. A. (1963). Semantical analysis of modal logic 1 normal modal propositional calculi. *Mathematical Logic Quarterly*, 9(5-6):67–96.

Krishnasamy Sivaprakasam, G. and Slutzki, G. (2016). Secrecy-preserving query answering in $\mathcal{ELH}$ knowledge bases. In *ICAART*.

Lutz, C., Sturm, H., Wolter, F., and Zakharyaschev, M. (2001). Tableaux for temporal description logic with constant domains. In *Automated Reasoning*, pages 121–136. Springer.

Tao, J., Slutzki, G., and Honavar, V. (2010). Secrecy-preserving query answering for instance checking in $\mathcal{EL}$. In *Proceedings of Web Reasoning and Rule Systems, 195–203*.

Tao, J., Slutzki, G., and Honavar, V. (2012). Pspace tableau algorithms for acyclic modalized $\mathcal{ALC}$. *Journal of Automated Reasoning*, 49(4):551–582.

Tao, J., Slutzki, G., and Honavar, V. (2014). A conceptual framework for secrecy-preserving reasoning in knowledge bases. *TOCL*, 16(1):3:1–3:32.

Tsukada, Y., Mano, K., Sakurada, H., and Kawabe, Y. (2009). Anonymity, privacy, onymity, and identity: A modal logic approach. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 3, pages 42–51. IEEE.