

# Privacy-Preserving Reasoning on the Semantic Web

Jie Bao, Giora Slutzki and Vasant Honavar

Department of Computer Science,  
Iowa State University,  
Ames, IA. USA. 50010  
{baojie,slutzki,honavar}@cs.iastate.edu

## Abstract

Many semantic web applications require selective sharing of ontologies between autonomous entities due to copyright, privacy or security concerns. In such cases, an agent might want to hide a part of its ontology while sharing the rest. However, prohibiting any use of the hidden part of the ontology in answering queries from other agents may be overly restrictive. We provide a framework for privacy-preserving reasoning in which an agent can safely answer queries against its knowledge base using inferences drawn based on both the hidden and visible part of the knowledge base, without revealing the hidden knowledge. We show an application of this framework in the widely used special case of hierarchical ontologies.

## 1 Introduction

The *semantic web* is aimed at making data, knowledge, and services in a form that allows software agents to find, share, integrate, and use data, knowledge, and services on the web. The imperative for sharing information in such a setting has to be balanced against copyright, privacy, security, or commercial concerns which require the participants to protect sensitive information from other parties. Hence, there is a need for mechanisms that enable the participants to selectively share information with other parties without risking disclosure of sensitive information. Our focus in this paper is on selective sharing of ontologies on the web: in particular, answering queries against an ontology, without disclosing *hidden knowledge*, i.e., knowledge that needs to be protected from disclosure.

Current proposals for *policy* languages [13] for information hiding on the semantic web rely on complete denial of access to the hidden parts of an ontology when answering queries against the ontology. We argue that such approaches are overly restrictive in that they prohibit the use of hidden knowledge in answering queries even in scenarios where it is possible to do so without disclosing the hidden knowledge.

Against this background, this paper examines the problem of answering queries against ontologies based on reasoning with hidden knowledge without risking its unintended disclosure. Specifically, we explore a framework for *privacy-preserving reasoning* on the semantic web. Unlike *Access control* policies used in databases [11] or their web counterparts (e.g., XACML [6]) that protect hidden knowledge on a *syntactic* level, our approach protects hidden knowledge on the *semantic* level. Thus, queries against an ontology can be answered based on inference using hidden knowledge whenever it is possible to do so without disclosing the hidden knowledge.

The main contributions of the paper are:

- A precise formulation of the problem of *privacy-preserving reasoning* on the semantic web.
- A general framework for privacy-preserving reasoning for semantic web ontologies that exploits the *indistinguishability* of hidden knowledge from *incomplete* knowledge under the *Open World Assumption* (OWA).
- A set of privacy-preserving reasoning strategies for *description logic* (DL) ontologies using the notion of *conservative extension* [8].
- Privacy-preserving reasoning strategies for the important special case of *hierarchical ontologies* via a reduction of privacy-preserving reasoning to *graph reachability* analysis.

## 2 Motivating Examples

We start with some examples of applications to motivate the need for privacy-preserving reasoning on the semantic web.

**Example 1 (Online Calendar):** Consider Bob who uses an online calendar to coordinate his daily activities with others. Suppose his calendar contains the fact that “Bob has a date with his girlfriend at noon”, along with additional knowledge, e.g., “date is a kind of activity” and “if person  $x$  has an activity at time  $t$ , person  $x$  is busy at time  $t$ ” and so on. Suppose Bob does not wish to share with his colleagues

details of his date. However, it might be necessary for his colleagues to know that Bob is busy at noon. In such a scenario, a query to Bob’s calendar as to whether Bob is busy at noon should be answered as “Yes” (which is inferred using both the sensitive knowledge and non-sensitive knowledge), whereas a query as to whether Bob has a date with his girlfriend at noon should be answered as “Unknown”. However, if the use of hidden knowledge was forbidden, it would be impossible for the calendar to inform Bob’s colleagues that “Bob is busy at noon”, although it is possible to do so, without revealing the details of Bob’s noon-time activity.

**Example 2 (Commercial Information Service):** Consider a company, say *U-Travel* that provides travel information to online customers. Suppose *U-Travel* offers a query service that provides limited information to the public but more detailed information to paying subscribers. Suppose the *U-Travel*’s ontology contains the following knowledge: (a) Sun Lodge is a 2-star hotel (b) a 2-star hotel is a hotel. Suppose *U-Travel* is willing to reveal that “Sun Lodge is a hotel” to the public, yet it wants to hide the fact that “Sun Lodge is a 2-star hotel” from all but its paying subscribers. If *U-Travel* query service could not use hidden information, i.e., that Sun Lodge is a 2-star hotel, it would not be able to inform a non-paying subscriber that Sun Lodge is a hotel, although it is possible to do so, without compromising hidden knowledge.

**Example 3 (Healthcare)** (based on a similar example given in [3]): Jane needs to take a certain preventive medicine for breast cancer. Suppose Jane does not want her physician or the pharmacy to supply the details of the prescription to her health insurance company, because she does not want to risk an increase in her health insurance premium on the basis of the fact that the medicine she has been prescribed is intended for use by women who are believed to have a high risk of developing breast cancer. In such a setting, in order for Jane to be reimbursed by her insurance company, the pharmacy needs to be able to certify to the insurance company (perhaps through a trusted third party) that Jane has indeed incurred a medical expense that is covered by her insurance policy.

One can easily imagine similar needs for selective sharing of inferences based on hidden knowledge in many other scenarios including, for example, business dealings between companies, interactions between different governmental agencies (e.g., intelligence, law enforcement, public policy), cooperation among independent nations on matters of global concern (e.g., counter-terrorism).

### 3 Partially Hidden Knowledge

We start by introducing the basic notion of hidden knowledge. A *knowledge base* (KB)<sup>1</sup>  $K$  over a language  $L$  consists of a set of *axioms*  $K = \{\alpha_1, \dots, \alpha_n\}$ . We assume that  $K$  is consistent and does not contain tautologies. We use  $\text{Sig}(\alpha_i)$  to denote the set of names occurring in an axiom  $\alpha_i$  and  $\text{Sig}(K)$  to denote the *signature* of a KB  $K$ ,  $\text{Sig}(K) = \cup_{i=1}^n \text{Sig}(\alpha_i)$ .

For a specific agent, the set of axioms that make up a KB  $K$  is divided into two mutually exclusive parts: a visible part  $K_v$  and a hidden part  $K_h$ , with the corresponding signatures  $\text{Sig}(K_v)$  and  $\text{Sig}(K_h)$ . We call  $\text{Sig}(K_v)$  the *visible signature* and  $\text{Sig}(K_h) - \text{Sig}(K_v)$  the *hidden signature*.

In what follows, a wide hat (e.g.,  $\widehat{\text{HiddenName}}$ ) is used to indicate that a name is hidden. We denote a KB  $K$  with a visible part  $K_v$  and a hidden part  $K_h$  by  $(K_v, K_h)$ .

We write  $K \vdash \gamma$  to mean that  $\gamma$  is *classically provable* from  $K$ . If every axiom in a KB  $K_2$  is classically provable from another KB  $K_1$ , we say that  $K_1$  *entails*  $K_2$  and denote it as  $K_1 \vdash K_2$ .

In some scenarios, it is useful to tailor the hidden and the visible parts of a KB  $K$  with respect to different agents that might query  $K$ . We call the division of the visible and the hidden KB of  $K$  w.r.t. an agent  $a$  the *scope policy* of  $K$  for agent  $a$ . In principle, a KB may have different scope policies for different agents. In what follows, we will focus on “safe” query answering for one agent against a partially hidden KB.

**Example 4:** Consider an ontology  $K = (K_v, K_h)$  of the *U-Travel* company. We use the *partial-order* relation  $\leq$  to indicate concept inclusion. The hidden part  $K_h$  contains

$$\begin{aligned} \text{SunLodge} &\leq \widehat{2\text{StarHotel}} \\ \widehat{2\text{StarHotel}} &\leq \text{Inn} \end{aligned}$$

where the hidden signature is  $\{\widehat{2\text{StarHotel}}\}$ . The visible part  $K_v$  contains

$$\begin{aligned} \text{SunLodge} &\leq \text{AAADiscountable} \\ \text{Inn} &\leq \text{Hotel} \end{aligned}$$

Hence, the visible signature is  $\text{Sig}(K_v) = \{\text{SunLodge}, \text{Inn}, \text{AAADiscountable}, \text{Hotel}\}$ .

### 4 Privacy-Preserving Reasoning

Our basic approach to designing a privacy-preserving reasoner for a partially-hidden KB is to ensure that the answers to queries do not inadvertently reveal hidden knowledge. The central idea is to design a reasoner that exploits

<sup>1</sup>In this paper, we use the terms “knowledge base” and “ontology” synonymously.

the *Open World Assumption* (OWA) of ontology languages, to make it impossible for the querying agent to distinguish between information that is unknown to the reasoner (because of the incompleteness of the KB) and the knowledge that is being protected by the reasoner. A query that cannot be safely answered without running the risk of disclosing hidden knowledge will be answered *as if* the reasoner lacks *the complete knowledge* to answer the query.

Unlike the Closed World Assumption (CWA) which is implicit in databases, OWA assumes that an ontology may be incomplete with regard to the knowledge of the world being modeled. Therefore, failure to prove an assertion does not imply the validity of the negation of the assertion. For instance, in Example 1, when queried whether “Bob has a date with his girlfriend at noon”, if the answer is “Unknown”, the querying agent cannot conclude that “Bob does not have such a date” (the negation of the assertion). Consequently, the querying agent cannot determine if the relevant information (the details of Bob’s noon-time activity) is not in the KB or if the information is in the KB but is protected.

Before we formalize the notion of privacy-preserving reasoning using hidden knowledge to answer queries against an ontology, we state some natural requirements that need to be met by a reasoner operating in the setting outlined above.

1. **Honesty.** The reasoner should not “lie”. That is, answers produced by the reasoner should always be *consistent* with its KB.
2. **History Independence.** The reasoner should always respond to a given query  $q$  against a fixed KB  $K$  with the same answer regardless of the *history* of queries that have been posed against  $K$ .
3. **History Safety.** The reasoner must ensure that the answers it produces are *safe* in the sense that it is not possible for a querying agent to infer any piece of hidden knowledge based on the answers to past queries from the same reasoner and the visible part of KB.

The first requirement is desirable if the goal of the reasoner is to provide as much information as it can, without providing wrong information (i.e., information that is inconsistent with its KB). The last two requirements are natural because it is unrealistic to assume that any reasoner that is used on the semantic web can “memorize” all previous queries that it has answered or track the identity of every agent that has queried it.

We now proceed to define a reasoner and a privacy-preserving reasoner:

**Definition 1 (Reasoner)** *Let  $K$  be a KB over a language  $L$ ,  $Q$  the query space (the set of possible assertions to be tested against  $K$ ) over  $L$ , and  $A$  the answer space. A reasoner  $R$  for  $K$  is an algorithm that defines a function  $R : K \times Q \rightarrow A$ . For a specific KB  $K$  we define  $R_K : Q \rightarrow A$  by setting  $R_K(q) = R(K, q)$ .*

An immediate consequence of this definition is that a reasoner  $R$  is “history independent” in the sense suggested by requirement 2 above.

$R$  might employ an *inference engine* which can be viewed as a classical reasoner with answer space  $A = \{Y, N\}$  such that  $\forall q \in Q, R_K(q) = Y$  iff  $K \vdash q$  (thus,  $R_K(q) = N$  iff  $K \not\vdash q$ ). While an inference engine always responds in a truthful manner, the reasoner, in order to protect some parts of  $K$ , may have an incentive to pick an answering strategy which does not respond with the “whole truth”. For example, a reasoner may answer “U” (Unknown) even if the correct answer (from the inference engine) is “Y” or “N”. The answer to a query  $q$  may be “U” either because the reasoner has incomplete knowledge (i.e.,  $K \not\vdash q$  and  $K \not\vdash \neg q$ ) under OWA, or because the “truthful” answer to  $q$  might risk disclosure of hidden knowledge. Under OWA, because the querying agent can not distinguish between these two cases, the reasoner is able to answer queries based on inference using hidden knowledge without revealing it.

**Definition 2 (Privacy-Preserving Reasoner)** *Let  $K = (K_v, K_h)$  be a KB over a language  $L$ ,  $Q$  the query space in  $L$ , and  $A = \{U, Y, N\}$  the answer space, and  $R$  a reasoner for  $K$ . We define:*

$$Q_Y = R_K^{-1}(Y), Q_N = R_K^{-1}(N), Q_U = R_K^{-1}(U).$$

*Clearly,  $Q = Q_U \cup Q_Y \cup Q_N$ . We require that  $q \in Q_Y$  iff  $\neg q \in Q_N$ .*

*(a)  $R$  is **strongly privacy-preserving** w.r.t.  $K$  if it satisfies the following two axioms:*

- **Honesty Axiom:**  $(q \in Q_Y \Rightarrow K \vdash q)$  and  $(q \in Q_N \Rightarrow K \vdash \neg q)$ .
- **Strong Safety Axiom:**  $\forall \alpha$  that is not a tautology and  $\text{Sig}(\alpha) \subseteq \text{Sig}(K_h), K_h \vdash \alpha \Rightarrow (K_v \cup Q_Y \not\vdash \alpha)$ .

*(b)  $R$  is **weakly privacy-preserving** w.r.t.  $K$  if it satisfies the Honesty Axiom and the following axiom:*

- **Weak Safety Axiom:**  $\forall \alpha, \alpha \in K_h \Rightarrow (K_v \cup Q_Y \not\vdash \alpha)$

The honesty axiom requires that reasoners provide answers that do not contradict the given KB (i.e.,  $K \cup Q_Y$  is consistent). The strong safety axiom requires that the answers provided by reasoners do not disclose any consequence that can be drawn from the hidden knowledge alone. The weak safety axiom requires the reasoner to protect only axioms (and their semantically equivalent syntactic variations) that are explicitly mentioned in the hidden part of the KB (but not necessarily their consequences).

The distinction between “strong safety” and “weak safety” is useful since different applications need different degrees of privacy preservation. In the *U-Travel* example, if the ontology provider is willing to disclose consequences of the hidden knowledge, e.g., “SunLodge  $\leq$  Inn”, it can get by with a weakly privacy-preserving reasoner. On the other

hand, in the online calendar example, suppose we have an additional piece of hidden knowledge “Alice is Bob’s girl friend”. Now, if Bob wants to protect *any* conclusion that may follow from the hidden part of his KB, e.g., that “Bob has a date with Alice at noon”, Bob will need a strongly privacy-preserving reasoner.

It can be shown that in a general setting, strong safety is a very restrictive requirement. For example, if there exist axioms  $\beta \in K_h$  and  $\gamma \in K_v$  (with  $\text{Sig}(\gamma) \subseteq \text{Sig}(K_h)$ ) such that  $\beta \vee \gamma$  is not a tautology, then there is no strongly privacy-preserving reasoner for  $K = (K_v, K_h)$ . On the other hand, weakly privacy-preserving reasoners exist for any KB that satisfies  $\alpha \in K_h \Rightarrow K_v \not\vdash \alpha$ . Intuitively, this means that no hidden axiom is provable from the visible KB. However, as we shall see in Section 6, it is possible to design strongly privacy-preserving reasoners in special cases, for instance, hierarchical ontologies (e.g., the *U-Travel* example).

## 5 Privacy-Preserving Reasoning: General Strategies

In this section, we discuss general strategies to designing privacy-preserving reasoners.

**Definition 3 (Strategy)** Let  $L$  be a language,  $\mathbf{K}_L$  the class of all knowledge bases over  $L$ , and  $\mathbf{R}_L$  the class of all reasoners over  $\mathbf{K}_L$ . A strategy for  $L$  is a function  $\mathfrak{R} : \mathbf{K}_L \rightarrow \mathbf{R}_L$  such that for every  $K \in \mathbf{K}_L$ ,  $R = \mathfrak{R}(K)$  is a reasoner for  $K$ . The strong/weak safety scope of a strategy  $\mathfrak{R}$ ,  $\text{Scope}(\mathfrak{R}) = \{K \in \mathbf{K}_L \mid \mathfrak{R}(K) \text{ is a strongly/weakly privacy-preserving reasoner for } K\}$ .

A strategy needs to compromise between two possibly conflicting goals:

1. *Generality*: An ideal strategy has the largest possible safety scope, i.e., is able to yield safe reasoners for the largest possible subclass of  $\mathbf{K}_L$ .
2. *Informativeness*: An ideal strategy is one that yields reasoners that provide as much information as possible in their answers to queries against their KBs, that is, reasoners that result in the smallest possible  $Q_U$ .

The following two strategies correspond to the “extreme” choices with respect to these two goals:

- *Dummy Strategy*, i.e., one that always generates a *dummy reasoner*, who answers “U” to every possible query against its KB. Obviously, a dummy reasoner is weakly privacy-preserving for any KB  $K = (K_v, K_h)$  such that  $\forall \alpha, \alpha \in K_h \Rightarrow K_v \not\vdash \alpha$ . Note that this condition is the weakest condition for a KB to have privacy-preserving reasoners. A dummy strategy is most general, but least informative. It has the largest scope, but answers given by reasoners that are based on it provide no information at all.

- *Naive Strategy*, i.e., one that generates a *naive reasoner* that reveals everything that follows from its knowledge base, including the hidden part of the KB. A naive reasoner is most informative, but is least general: It is privacy-preserving only for those KB that have no hidden knowledge at all (i.e.,  $K_h = \emptyset$ ).

In practice, we may need to make tradeoff between the conflicting requirements of generality and informativeness of strategies.

We now proceed to present a general approach for generating weakly privacy-preserving reasoners for semantic web ontologies based on the notion of *conservative extension* [8]. The basic idea behind this approach is as follows: Answers to previous queries may be used by the querying agent to *extend* the visible part of the KB. The safety of the strategy can be guaranteed if we can ensure that no conclusions compromising the hidden knowledge can be inferred from such an extension.

**Definition 4 (Conservative Extension, [8])** Let  $K$  and  $K'$  be two knowledge bases.  $K \cup K'$  is a conservative extension of  $K$ , written as  $K \cup K' \rightleftharpoons K$ , if for every formula  $\alpha$  such that  $\text{Sig}(\alpha) \subseteq \text{Sig}(K)$ ,  $K \cup K' \vdash \alpha$  iff  $K \vdash \alpha$ .

Let  $K_{vc} \subseteq K_v$  be the set of visible axioms that contain names in  $\text{Sig}(K_h)$ . Hence,  $\text{Sig}(K_v) \cap \text{Sig}(K_h) \subseteq \text{Sig}(K_{vc})$ . Intuitively, because the querying agent does not know names that are not in  $\text{Sig}(K_v)$ , the names in  $K_{vc}$  correspond to “critical signature”, i.e., the subset of  $\text{Sig}(K_h)$  that is known to the querying agent. If we can ensure that answers to queries together with  $K_v - K_{vc}$  do not reveal any names that belong  $\text{Sig}(K_h)$  beyond those in  $\text{Sig}(K_{vc})$ , we can effectively protect every axiom in  $K_h$ . Therefore, if we can ensure that any extension of  $K_v$  with the results of previous queries is a conservative extension of the *critical* visible axioms  $K_{vc}$ , we can protect hidden knowledge. The following lemma captures this intuition more formally:

**Lemma 1** Let  $K = \{K_v, K_h\}$  be a KB such that  $\forall \alpha, \alpha \in K_h \Rightarrow K_v \not\vdash \alpha$ ,  $R$  a reasoner for  $K$ .  $R$  is a weakly privacy-preserving reasoner for  $K$  if it satisfies the honesty axiom and  $K_v \cup Q_Y \rightleftharpoons K_{vc}$ .

*Proof*: We only need to show that the weak safety axiom holds under the stated conditions:

$$\begin{aligned} \alpha \in K_h &\Rightarrow K_v \not\vdash \alpha \\ &\Rightarrow K_{vc} \not\vdash \alpha \\ &\Rightarrow K_v \cup Q_Y \not\vdash \alpha \quad \square \end{aligned}$$

## 6 Privacy-Preserving Reasoning with SHIQ Ontologies

In this section we present a “safe” reasoning strategy based on conservative extensions for the description logic

*SHIQ*, which covers a significant part of OWL. Grau et al. [7] have shown that in the special case of *semantically local* ontologies (see below), it is possible to check whether an extension of a *SHIQ* ontology is a conservative extension in polynomial time. Informally, an axiom is *semantically local* w.r.t. a signature  $S$  if it imposes no restrictions on the interpretation of names in  $S$ . A finite set of axioms is local w.r.t.  $S$  if every axiom in it is local w.r.t.  $S$ . Practical ways to ensure *semantic locality* of *SHIQ* ontologies have been elucidated by Grau et al. [7]. They have also established relationship between the notions of conservative extension and semantic locality of ontologies which we summarize (adapted for simpler presentation in our setting) in the following lemma:

**Lemma 2** (Definition 3 and Lemma 5 of [7]) *Suppose  $K_1$  and  $K_2$  are two SHIQ TBoxes such that  $K_1$  is local w.r.t.  $\text{Sig}(K_2)$  and  $K_2$  is local w.r.t.  $\emptyset$ . Then  $K_1 \cup K_2$  is a conservative extension of  $K_2$ .*

We can now define  $\mathfrak{R}_{CE}$  (read *CE-strategy*), a reasoning strategy for *SHIQ* ontologies, based on the notion of conservative extension. Given a *SHIQ* TBox  $K = \{K_v, K_h\}$  and subsumption query  $q$ ,  $\mathfrak{R}_{CE}$  specifies a reasoner for  $K$  that answers  $q$  as follows<sup>2</sup>

```

IF  $q$  is local w.r.t.  $\text{Sig}(K_{vc})$  and  $\text{Sig}(q) \subseteq \text{Sig}(K_v)$ 
  IF  $K \vdash q$ , return  $Y$ 
  ELSE IF  $K \vdash \neg q$ , return  $N$ 
    ELSE return  $U$  /*incomplete knowledge*/
  ELSE return  $U$  /*hidden knowledge*/

```

**Lemma 3** *The weak safety scope of  $\mathfrak{R}_{CE}$  includes all SHIQ TBoxes  $K = \{K_v, K_h\}$  that satisfy the following properties:*

- $K_v - K_{vc}$  is local w.r.t.  $\text{Sig}(K_{vc})$ ;
- $K_{vc}$  is local w.r.t.  $\emptyset$ ;
- $\forall \alpha \in K_h, K_v \not\vdash \alpha$

*Proof:* Let  $R = \mathfrak{R}_{CE}(K)$ , where  $K$  satisfies the given properties. Clearly,  $R$  satisfies the honesty axiom. By the definition of the algorithm,  $Q_Y$  is local w.r.t.  $\text{Sig}(K_{vc})$ . Since  $K_v - K_{vc}$  is local w.r.t.  $\text{Sig}(K_{vc})$ ,  $Q_Y \cup (K_v - K_{vc})$  is local w.r.t.  $\text{Sig}(K_{vc})$ . Since  $K_{vc}$  is local w.r.t.  $\emptyset$ , by Lemma 2,  $(K_v - K_{vc}) \cup Q_Y \cup K_{vc} = K_v \cup Q_Y \rightleftharpoons K_{vc}$ . By Lemma 1,  $R$  is a weakly privacy-preserving reasoner for  $K$ .  $\square$

It is worth noting that we do not require the hidden knowledge  $K_h$  also to be local, as long as  $\alpha \in K_h \Rightarrow K_v \not\vdash \alpha$ . This affords considerable flexibility for ontology engineers in designing the KB.

<sup>2</sup>Note that if  $q := C \sqsubseteq D$ ,  $K \models q$  means that for every model  $\mathcal{I}$  of  $K$ ,  $(C \sqcap \neg D)^{\mathcal{I}} = \emptyset$ ; thus  $K \models \neg q$  means that for every model  $\mathcal{I}$  of  $K$ ,  $(C \sqcap \neg D)^{\mathcal{I}} \neq \emptyset$ . Also note that since there are sound and complete reasoners for *SHIQ*,  $K \models q$  iff  $K \vdash q$ .

An important advantage of the *CE-strategy* for *SHIQ* ontologies is that a weakly privacy-preserving reasoner can be built as a meta reasoner which calls inference service from a standard sound and complete DL reasoner for *SHIQ*. Thus, implementing a weakly privacy-preserving reasoners for *SHIQ* ontologies is quite straightforward.

## 7 Privacy-Preserving Reasoning with Hierarchical Ontologies

Unlike in the case of general DL ontologies, it is possible to define a strongly safe strategy, and hence strongly safe privacy-preserving reasoners in the case of *hierarchical* ontologies, e.g., tree or DAG-structured ontologies.

Formally, a hierarchical ontology  $K$  over a finite set of names  $S$  can be represented as a set of visible or hidden partial order axioms, denoted by  $K = (S, \leq)$  (as illustrated in Example 4).

Reasoning with  $K$  can be reduced to the graph reachability problem by defining a corresponding directed graph  $G = (V, E)$ , where  $V$  is the vertex set corresponding to elements of  $S$ , and  $E$  is the edge set corresponding to  $\leq$  axioms. Let  $\mathbf{G}$  be the set of all directed graphs. In the following, we will identify  $K$  with the corresponding  $G$ .

A vertex (or edge) is said to be a visible vertex (or edge) if it is mapped from a visible term (or axiom); otherwise it is said to be hidden. Let  $E_h$  be the set of all hidden edges,  $E_v$  be the set of all visible edges, and  $E = E_h \cup E_v$ . For any set edges  $F \subseteq E$ , let  $F^+$  denote the transitive closure of  $F$ , and  $F^{\leq m} = \cup_{k=1}^m F^k$ .

For any two visible vertices  $x$  and  $y$ ,  $y \leq x$  if  $x$  is reachable from  $y$  in the graph  $G$ , i.e., there exists a path from  $y$  to  $x$  (which in general, may contain both visible and hidden edges). Note that because of the open world assumption, it is not necessarily the case that  $y \leq x$  is false simply because there is no path from  $y$  to  $x$  in  $G$ .

An affirmative answer to a query about the reachability from  $y$  to  $x$  in  $G$  is equivalent to augmenting  $G$  by adding a new visible edge  $\langle y, x \rangle$ . Hence, in order to realize privacy-preserving reasoning, we should ensure that the initial graph  $G$  (derived from  $K$ ) can be augmented with previous answers without revealing the existence of hidden edges.

First, it is easy to see that strong safety and weak safety properties can be reduced to each other in the case of hierarchical ontologies:

**Lemma 4**  *$R$  is a strongly privacy-preserving reasoner for  $G = (V, E_v \cup E_h)$  iff  $R$  is a weakly privacy-preserving reasoner for  $G = (V, E_v \cup E_h^+)$ .*

This lemma follows from the fact that  $E_h^+$  contains all possible inference results that can be obtained by considering *only* the hidden edges  $E_h$ . Henceforth, we will focus on weak safety.

We now proceed to define several classes of graphs that have safe strategies with different degree of informativeness:

$$\mathbf{S}_{m,n} = \{G \in \mathbf{G} \mid (E^{\leq m} - E_h)^{\leq n} \cap E_h = \emptyset\},$$

where  $m, n$  can be “+” indicating transitive closure. Graphs in  $\mathbf{S}_{m,n}$  are called  $(m, n)$ -safe. Intuitively,  $m$  represents the ability of the reasoner to detect possible safety hazards, and  $n$  represents the ability of the querying agent to discover knowledge from previous answers and the visible part of the graph. We are only interested in the case when  $n$  is + since it represents the case when the querying agent is the most powerful<sup>3</sup>. It is easy to verify that:  $\mathbf{S}_{+,+} = \bigcap_{m=1}^{\infty} \mathbf{S}_{m,+}$ .

Now we will consider several specific reasoning strategies for those classes of graphs. Since there is no negation, the resulting reasoners will answer only with “Y” or “U” (i.e., there are no “N” answers). Note that requirements for a weakly privacy-preserving reasoner in this context are:

- **Honesty Axiom:**  $Q_Y \subseteq E^+$
- **Weak Safety Axiom:**  $(E_v \cup Q_Y)^+ \cap E_h = \emptyset$ .

**The dummy reasoner:** A dummy reasoner responds to every query with the answer “U” (i.e.,  $Q_Y = \emptyset$ ). It preserves the safety of precisely those graphs that satisfy  $E_v^+ \cap E_h = \emptyset$ . This is exactly the defining condition of the class of  $(1,+)$ -safe graphs  $\mathbf{S}_{1,+}$ . Obviously, this strategy has the widest safety scope, and not surprisingly, is also least informative.

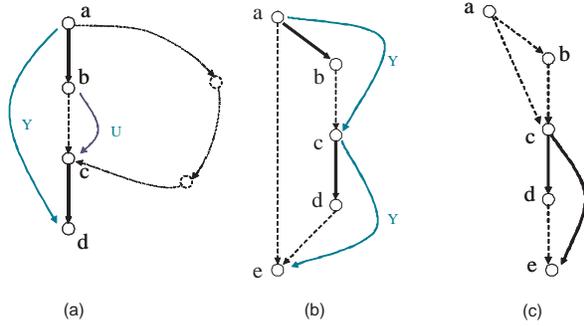
**The obvious reasoner:** An obvious reasoner responds with an answer “Y” to only those queries whose answers follow from  $E_v$  i.e.  $Q_Y \subseteq E_v^+$ . Its weak safety scope is the same as that of the dummy reasoner ( $\mathbf{S}_{1,+}$ ).

**The safe reasoner:** let  $Q_Y = E^+ - E_h$ . Clearly, this reasoner satisfies the honesty axiom. To satisfy the weak safety axiom, we need  $(E_v \cup (E^+ - E_h))^+ \cap E_h = \emptyset$ , i.e.,  $(E^+ - E_h)^+ \cap E_h = \emptyset$ . Hence, its weak safety scope is  $\mathbf{S}_{+,+}$  (which is smaller than  $\mathbf{S}_{1,+}$ ). Similarly, if  $Q_Y = E^{\leq m} - E_h$ , the corresponding weak safety scope is  $\mathbf{S}_{m,+}$ .

**The naive reasoner:** this reasoner always gives away all the information it has, i.e.,  $Q_Y = E^+$ . It is trivially honest, and its weak safety scope consists of only those graphs that do not have any hidden edges, i.e.,  $\{G \mid E_h = \emptyset\}$ , which is clearly a subset of  $\mathbf{S}_{+,+}$ .

The above reasoning strategies for hierarchical ontologies clearly illustrate the tradeoff between generality and informativeness. Among these, the *safe* reasoner is of particular interest since it is able to generate informative answers for a fairly large class of graphs without compromising the

hidden knowledge<sup>4</sup>. Given a query  $x \leq y$ , it will answer “Y” if  $\langle x, y \rangle$  is in  $E^+$  but *not* in  $E_h$ , and “U” otherwise.



Solid edges are visible, dashed edges are hidden, edges labeled with “Y” or “U” are queries with corresponding answers, dotted edges in (a) show some possible edges that are not present in the graph.

### Figure 1. Safety of Hierarchical Ontologies

Some examples of hierarchical ontologies are shown in Figure 1 as graphs. The graph in (a) is  $(+, +)$ -safe. For instance, the query  $a \leq d$  can be answered “Y” using both visible and hidden edges without revealing hidden edge  $b \leq c$ . This is because, due to OWA, there might exist unknown paths that connect  $a$  and  $d$  but not  $b$  and  $c$ . The graph shown in (b) is not  $(+, +)$ -safe: two “seemingly safe” queries  $a \leq c$  and  $c \leq e$  may be combined to reveal the hidden edge  $a \leq e$ . Note that the graph will become  $(+, +)$ -safe if we remove the hidden edge  $a \leq e$ . It is easy to verify that the graphs in Figure 1 (c) and Example 4 are also  $(+, +)$ -safe.

## 8 Related Work

Problems of trust and privacy on the web in general, and the semantic web in particular, are topics of significant current interest. For example, research on policy languages [13, 1, 12] focus on mechanisms for controlling the access to resources or operations. Research on encryption of sensitive information focuses on preventing unauthorized access to such information using cryptographic protocols. W3C XML Encryption<sup>5</sup> working group has proposed an XML syntax for encrypting or decrypting digital content in XML documents. Giereth [4] has studied the hiding of a fragment of an RDF document by encrypting it while the rest of the document remains publicly readable.

These access control policies and encryption techniques either allow or prohibit access to sensitive information: They do not allow the use of private knowledge to answer queries that can be safely answered using private knowledge

<sup>3</sup>We note that for every  $m, n$ , we can easily construct a reasoning strategy whose safety scope is  $\mathbf{S}_{m,n}$ .

<sup>4</sup>A simple Java implementation of the reasoner is available at <http://www.cs.iastate.edu/~baojie/pub/wi2007>

<sup>5</sup><http://www.w3.org/Encryption/2001/>

without revealing it. In contrast, this paper explores how to relax the restriction on access to private or hidden knowledge so that they can still be used in answering queries with guaranteed safety.

Similar in spirit to our work is the work of Farkas et al. [3, 10] on preventing unwanted inferences in data repositories. Their *privacy information flow model* [3] includes a privacy mediator that prevents agents in the system from being able to (indirectly) infer the private data of other agents. The inference algorithm assumes a tree-like data model, selection-projection queries, and a domain knowledge base (consisting of assertions in the form of Horn clauses) with closed world semantics. In contrast, the approach proposed in this paper assumes open world semantics which is more natural in the semantic web setting.

Jain and Farkas [10] have proposed an RDF authorization model that can selectively control access to stored RDF triples, assign security classification to inferred RDF triples and check for unauthorized inferences. This model assigns a highest-level security label to each (stored or inferred) RDF triple using a pre-specified set of syntactic rules. In contrast, the focus of our work is on mechanisms that allow the use of hidden knowledge to answer queries without revealing hidden knowledge.

Logics with specially designed semantics have also been applied in information hiding. Cuppens [2], Glasgow et al. [5] describe a logic that includes epistemic operators for reasoning about knowledge and deontic operators for reasoning about permission and obligation. Gray and Syverson [9] adopt a hybrid approach by connecting the information-theoretic formulations of security and logical formulations of knowledge and probability in distributed systems. In contrast, our approach is based on the standard first-order semantics, thus is easier to implement based on existing reasoners.

## 9 Conclusion

We have presented a framework that allows knowledge to be hidden but not forbidden from *safe* use in answering queries. We have taken some initial steps towards identifying the requirements of privacy-preserving reasoning and developing privacy-preserving reasoners for description logics and hierarchical ontologies. A privacy-preserving reasoner in our setting can be realized as a meta reasoner that operates on top of existing reasoners.

We have shown how to design strongly privacy preserving reasoners in the special case of hierarchical ontologies.

Some directions for future work include investigation of strongly privacy-preserving reasoners for more expressive ontology languages. Our immediate goal is to extend the graph theory based safe reasoning strategy to more expressive graphs, e.g. RDF.

The discussion in this paper has been limited to privacy-preserving reasoning over a single knowledge base. It would be interesting to explore privacy-preserving reasoning in multi-agent setting over several knowledge bases, each controlled by an autonomous agent (and selectively shared with other agents).

**Acknowledgement:** This research was supported in part by the US NSF award 0639230

## References

- [1] P. A. Bonatti, C. Duma, N. Fuchs, W. Nejdl, D. Olmedilla, J. Peer, and N. Shahmehri. Semantic web policies - a discussion of requirements and research issues. In *ESWC*, pages 712–724, 2006.
- [2] F. Cuppens. An epistemic and deontic logic for reasoning about computer security. In *First European Symposium On Research In Computer Security (ESORICS 90)*, pages 135–145, 1990.
- [3] C. Farkas, A. Brodsky, and S. Jajodia. Unauthorized inferences in semi-structured databases. *Information Sciences*, 176(22):3269–3299, Nov. 2006.
- [4] M. Giereth. On partial encryption of rdf-graphs. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2005.
- [5] J. I. Glasgow, G. H. MacEwen, and P. Panangaden. A logic for reasoning about security. *ACM Trans. Comput. Syst.*, 10(3):226–264, 1992.
- [6] S. Godik and T. Moses. Oasis extensible access control markup language (xacml). OASIS Committee Specification cs-xacml-specification-1.0, November 2002, <http://www.oasis-open.org/committees/xacml/>, 2002.
- [7] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. A logical framework for modularity of ontologies. In *IJCAI*, pages 298–303, 2007.
- [8] B. C. Grau, I. Horrocks, O. Kutz, and U. Sattler. Will my ontologies fit together? In *Proc. of the 2006 Description Logic Workshop (DL 2006)*, volume 189. CEUR (<http://ceur-ws.org/>), 2006.
- [9] J. W. Gray and P. F. Syverson. A logical approach to multi-level security of probabilistic systems. *Distributed Computing*, 11(2):73–90, 1998.
- [10] A. Jain and C. Farkas. Secure resource description framework: an access control model. In *SACMAT*, pages 121–129, 2006.
- [11] S. Jajodia and D. Wijesekera. Recent advances in access control models. In *DBSec*, pages 3–15, 2001.
- [12] L. Kagal, M. Paolucci, N. Srinivasan, G. Denker, T. W. Finin, and K. P. Sycara. Authorization and privacy for semantic web services. *IEEE Intelligent Systems*, 19(4):50–56, 2004.
- [13] G. Tonti, J. M. Bradshaw, R. Jeffers, R. Montanari, N. Suri, and A. Uszok. Semantic web languages for policy representation and reasoning: A comparison of kaos, rei, and ponder. In D. Fensel, K. P. Sycara, and J. Mylopoulos, editors, *International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 419–437. Springer, 2003.