Secrecy-Preserving Reasoning Over Entailment Systems
Theory and Applications

George Voutsadakis, Giora Slutzki, Vasant Honavar
Department of Computer Science Iowa State University Ames, IA 50011, USA

Abstract

Privacy, copyright, security and other concerns make it essential for many distributed web applications to support selective sharing of information while, at the same time, protecting sensitive knowledge. Secrecy-preserving reasoning *refers to the answering of queries against a knowledge base involving inference that uses sensitive knowledge without revealing it. We present a general framework for secrecy-preserving reasoning over arbitrary* entailment systems. *This framework enables reasoning with hierarchical ontologies, propositional logic knowledge bases (over arbitrary logics) and RDFS knowledge bases containing sensitive information that needs to be protected. We provide an algorithm that, given a knowledge base over an effectively enumerable entailment system, and a secrecy set over it, defines a maximally informative secrecy-preserving reasoner.* Secrecy-preserving mappings *between knowledge bases that allow reusing reasoners across knowledge bases are introduced.*

1. Introduction

Problems of trust, privacy and security in information systems in general, and networked information systems (e.g., the web), in particular, are topics of significant current interest. In many applications in the semantic web, sharing of information is of utmost importance. This need has to be balanced against the competing requirement to protect sensitive or confidential information from unintended disclosure.

Example 1: Suppose that John buys Drug A for cancer. Drug A is a generic drug and generic drugs are covered by John's insurance policy. Suppose that the exact drug that John takes is to be kept secret from his insurance company lest his condition becomes known and, as a result, his insurance premiums are unreasonably increased. If the knowledge

Drug A is a generic drug
Generic drugs are covered by insurance policy

is combined with the secret knowledge

John buys Drug A

the information

John is covered by insurance policy,

needed for reimbursement, can be inferred without disclosing the secret knowledge. □

One can easily imagine similar needs for selective sharing of results of inference based on protected knowledge in many other scenarios including, for example, interactions among business partners, different governmental agencies (e.g., intelligence, law enforcement, public policy), or independent nations acting on matters of global concern (e.g., counter-terrorism). The focus of this paper is on applications where there is a need to reason and deduce new information from existing knowledge bases, in which part of the knowledge needs to be protected.

Early work on securing information focused on access control mechanisms (see [2] for a survey). For, instance, work on *policy languages* for the web [3], [12] involves specifying syntax-based restrictions on access to specific resources or operations on the web. Giereth [8] has studied the hiding of a fragment of an RDF document by encrypting it while the rest of the document remains publicly readable. Farkas et al. [5], [11] have proposed a *privacy information flow model* to prevent unwanted inferences in data repositories. Jain and Farkas [11] have proposed an RDF authorization model that can selectively control access to stored RDF triples using a pre-specified set of *syntactic* rules. In a recent paper [4] Grau and Horrocks have introduced a framework that combines logic and probabilistic approaches to guarantee privacy preservation.

Most of the existing approaches to the protection of secret information rely on forbidding access to the sensitive parts of a knowledge base. Such approaches can be overly restrictive in scenarios where it is possible, and may be desirable, for a knowledge base to use both secret and publicly available knowledge to answer queries without risking disclosure of the secret knowledge [1]. Such reasoning was termed *privacy-preserving reasoning*. A precise formulation of the problem of privacy-preserving reasoning was provided in [1] and a framework was developed to tackle the problem based on the Open World Assumption (OWA).

In this paper, we introduce *secrecy-preserving reasoning* over arbitrary entailment systems that extends the notion of privacy-preserving reasoning to handle a broader gamut of applications. Our goal is to devise a very general framework that is flexible enough to cover a wide variety of real-world knowledge bases and is easily adaptable to various concrete application scenarios. Once the framework is presented, to illustrate its usefulness, we apply it to hierarchical knowledge bases, to knowledge bases over arbitrary propositional languages, to hypergraphical knowledge bases and, finally, to RDFS knowledge bases. The latter encompass many of the knowledge bases that are currently available in the semantic web.

The main contributions of the present paper are presented next.

- We introduce a very general framework for secrecy-preserving reasoning with arbitrary entailment systems.
- We present an algorithm that, given a knowledge base with sensitive information and a linear ordering of the set of possible queries, devises a secrecy-preserving reasoner for answering queries against the knowledge base. The resulting reasoner is maximal in the sense that it reveals as much information as possible without risking disclosure of the secret information.
- We introduce the concept of secrecy-preserving mapping between ontologies containing sensitive information. Secrecy-preserving mappings help in reducing reasoning in one ontology to reasoning in another, while maintaining secrecy features, leading to more effective reuse of ontologies.
- We apply this framework to secrecy-preserving reasoning with RDFS knowledge bases.

The rest of the paper is organized as follows: Section 2 reviews the general framework for secrecy-preserving reasoning. Section 3 describes secrecy-preserving reasoning over arbitrary entailment systems and provides several examples. Section 4 presents the general algorithm, which, given a knowledge base with sensitive information, outputs a secrecy-preserving reasoner for answering queries against the knowledge base. It also states various properties of this algorithm. Subsection 4.2 looks more closely at hierarchical ontologies and shows how the general algorithm is applied to this special case. Section 5 introduces and studies secrecy-preserving mappings between entailment systems that enable structural comparisons between ontologies with hidden components and facilitate the reuse of secrecy-preserving reasoners. Section 6 shows, as an application, how we can use the general framework to perform secrecy-preserving reasoning with RDFS knowledge bases. Finally, Section 7 concludes with a summary.

2. Secrecy-preserving Reasoning: General Framework

Let $\mathcal{L}$ be a signature or language type, i.e., a set of connectives with predetermined arities. Associated with $\mathcal{L}$ is a set of formulas $\mathrm{Fm}_{\mathcal{L}}$, which are built using the connectives in $\mathcal{L}$ starting from a denumerable set of atomic propositions (in the case of a propositional logic) or names (in the case of a description logic) in the ordinary recursive way.

An **inference system** $\mathcal{S} = \langle \mathcal{L}, \vdash_{\mathcal{S}} \rangle$ consists of a language type $\mathcal{L}$ together with a finitary consequence relation $\vdash_{\mathcal{S}} \subseteq \mathcal{P}(\mathrm{Fm}_{\mathcal{L}}) \times \mathrm{Fm}_{\mathcal{L}}$ on the set of formulas over $\mathcal{L}$, where $\mathcal{P}(X)$ denotes the powerset of a set X. Given $\Phi \subseteq \mathrm{Fm}_{\mathcal{L}}$, we use the notation $\Phi^{+} = \{\phi \in \mathrm{Fm}_{\mathcal{L}} : \Phi \vdash_{\mathcal{S}} \phi\}$ to denote the set of $\mathcal{S}$-**consequences** of Φ. Since $\mathcal{S}$ will be fixed, this will not cause any confusion. The **interderivability relation** associated with $\mathcal{S}$ is the relation

$$\Lambda(\mathcal{S}) = \{\langle \phi, \psi \rangle \in \mathrm{Fm}_{\mathcal{L}}^2 : \{\phi\}^{+} = \{\psi\}^{+}\},$$

i.e., it is the relation that contains all pairs of $\mathcal{S}$-interderivable formulas[1].

A **knowledge base** $\mathbf{K}$ over $\mathcal{S}$ is a quadruple $\mathbf{K} = \langle K, B, Q, A \rangle$, where

- K is a finite consistent set of formulas;
- $B \subseteq K$ is the **browsable part** of K, i.e., the part that can be accessed by any querying agent without any restrictions. If $B = \emptyset$, then access to the knowledge base is possible only via queries.
- $Q \subseteq \mathrm{Fm}_{\mathcal{L}}$ is the **query space** and it satisfies $K^{+} \subseteq Q$. It represents the set of all queries that can be posed against $\mathbf{K}$. We stipulate that
    1) If $\langle \phi, \psi \rangle \in \Lambda(\mathcal{S})$ and $\phi \in Q$, then $\psi \in Q$, i.e., that the interderivability relation of $\mathcal{S}$ is *compatible with* Q in the sense of [6]. Note that this means that Q is a union of equivalence classes of $\Lambda(\mathcal{S})$;
    2) If the language includes negation, Q is closed under negation.

1. This is called the **Frege relation** of $\mathcal{S}$ in [6].

- A is the **answer space**, i.e., the set of all possible answers that the knowledge base may return to a query. In most examples, we will have $A = \{Y, N, U\}$ or $A = \{Y, U\}$, where Y, N, U stand, respectively, for YES, NO and UNKNOWN.

Intuitively, if $\phi \in Q$, then posing the query ϕ is tantamount to asking whether $\phi \in K^+$.

From now on, unless stated otherwise, we will fix the answer set to be the set $A = \{Y, N, U\}$. Consider the knowledge base $\mathbf{K} = \langle K, B, Q, A\rangle$. For any function $R : Q \to A$, define

$$Q_Y = R^{-1}(Y) \quad Q_N = R^{-1}(N) \quad Q_U = R^{-1}(U).$$

The sets Q_Y, Q_N, Q_U obviously form a partition of the query set Q.

A **reasoner** for $\mathbf{K}$ or a $\mathbf{K}$**-reasoner** is a computable function $R : Q \to A$, satisfying the following axioms:

- **Invariance:** If $\langle\phi, \psi\rangle \in \Lambda(\mathcal{S})$, then $R(\phi) = R(\psi)$, for all $\phi, \psi \in Q$;
- **Yes-Axiom:** $B^+ \subseteq Q_Y \subseteq K^+$;
- **No-Axiom:** If the language has negation, $Q_N = \neg Q_Y$.

Note that, if $\mathbf{K}$ is a knowledge base and R a $\mathbf{K}$-reasoner, then

$$Q_N \cap K^+ = \emptyset \quad \text{and} \quad Q_U = \neg Q_U,$$

the latter holding whenever $\mathcal{L}$ has negation.

Next, we explore a canonical method that may be used to weaken the informativeness of a given reasoner without affecting any of its key properties. Expressed differently, given a reasoner for a knowledge base $\mathbf{K}$, we provide a method for creating a variety of less informative reasoners.

Proposition 1: Let $\mathbf{K} = \langle K, B, Q, A\rangle$ be a knowledge base over an inference system $\mathcal{S}$ with negation and $R : Q \to A$ a $\mathbf{K}$-reasoner. Suppose $W \subseteq Q_Y$, such that

1) $W \cap B^+ = \emptyset$;
2) $\Lambda(\mathcal{S})$ is compatible with each of $W, \neg W$, $Q_Y \backslash W$ and $Q_N \backslash \neg W$.

Then, the function $R' : Q \to A$, defined, for all $\phi \in Q$, by

$$R'(\phi) = \begin{cases} Y, & \text{if } \phi \in Q_Y \backslash W \\ N, & \text{if } \phi \in Q_N \backslash \neg W \\ U, & \text{if } \phi \in Q_U \cup (W \cup \neg W) \end{cases}$$

is also a $\mathbf{K}$-reasoner.

Proof: One has to check that all three properties in the definition of a reasoner are satisfied by the new function $R' : Q \to A$.

- For Invariance, suppose that $\phi, \psi \in \mathrm{Fm}_{\mathcal{L}}$, such that $\langle\phi, \psi\rangle \in \Lambda(\mathcal{S})$. If $R'(\phi) = Y$, then $\phi \in Q_Y \backslash W$. Thus, since $\Lambda(\mathcal{S})$ is compatible with $Q_Y \backslash W$, we get that $\psi \in Q_Y \backslash W$, which implies that $R'(\psi) = Y$. The case $R'(\phi) = N$ is handled similarly. Finally, if $R'(\phi) = U$, then $\phi \in Q_U \cup W \cup \neg W$. If $\phi \in Q_U$, then $\psi \in Q_U$. If $\phi \in W$ or $\phi \in \neg W$, then the compatibility of $\Lambda(\mathcal{S})$ with W and with $\neg W$ implies that $\psi \in W$ or $\psi \in \neg W$, respectively. Therefore $R'(\psi) = U$, as well. This shows that R' is invariant.
- For the Yes-Axiom, we have

$$B^+ \subseteq Q_Y \backslash W = Q'_Y \subseteq Q_Y \subseteq K^+.$$

- That the No-Axiom holds follows directly from the definition of R'.

□

Suppose now that the knowledge base $\mathbf{K}$ is queried by a querying agent that has at its disposal an inference engine for the inference system $\mathcal{S}$. Suppose, also, that the knowledge base is required to keep secret a set $S \subseteq K^+ \backslash B^+$. The set S will be termed the **secret set** or **secrecy set**. Note that information in the browsable part B is readily available and, therefore, all its consequences can be inferred by the querying agent. Thus, no information in B^+ can be kept confidential. To protect the information contained in S, a $\mathbf{K}$-reasoner must answer U to any query in S, i.e., we must have $S \subseteq Q_U$. This, however, may not be enough because it may be the case that knowledge outside S can be used to deduce information in S. This knowledge would also have to be concealed by $\mathbf{K}$. Thus, a **secrecy-preserving $\mathbf{K}$-reasoner for S** must specify a set E_S, called a **secrecy envelope** or **security envelope of** S, such that

- **Enveloping Axiom:** $S \subseteq E_S \subseteq K^+ \backslash B^+$;
- **Secrecy Axiom:** $(K^+ \backslash E_S)^+ \cap S = \emptyset$.

Once such a set is fixed the secrecy-preserving $\mathbf{K}$-reasoner for S (associated with E_S) is the function $R : Q \to A$ defined, for all $\phi \in Q$, by

$$R(\phi) = \begin{cases} Y, & \text{if } \phi \in K^+ \backslash E_S \\ N, & \text{if } \neg\phi \in K^+ \backslash E_S \\ U, & \text{otherwise} \end{cases} \tag{1}$$

A few remarks concerning a secrecy envelope of a secrecy set S are in order. This set is *not unique*, given the secrecy set S. However, the goal is to keep it as small as possible. The reason is that, as seen by Equation (1), the smaller the secrecy envelope of S is, the larger the set $Q_Y \cup Q_N$ that results (i.e., the smaller the set Q_U) and, hence, the *more informative* the $\mathbf{K}$-reasoner obtained via Equation (1).

Our basic approach to designing secrecy-preserving reasoners for KBs that contain sensitive knowledge is to ensure that the answers to queries do not reveal information in the secrecy set. The underlying idea is to design a reasoner that exploits 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 secret knowledge will be answered *as if* the reasoner *lacks the complete knowledge* to answer the query.

In the next section, we generalize this framework to secrecy-preserving reasoning over arbitrary entailment systems. This general setting is adopted with an eye towards a unified treatment of a wide range of applications calling for secrecy-preserving reasoning. Its noteworthy feature is that it scales down in an elegant way to diverse secrecy-preserving reasoning needs that arise in practice. Thus, it makes it unnecessary for a user to devise a new framework each time a new application arises.

3. Secrecy-preserving Reasoning Over Entailment Systems

Let X be an arbitrary set. A **rule of inference over** X is a pair $\langle Y, y\rangle$, where Y is a finite subset of X and $y \in X$. Sometimes, the notation $\frac{Y}{y}$ will be used in place of $\langle Y, y\rangle$.

An **entailment system** is a pair $\mathcal{E} = \langle X, \mathcal{R}\rangle$ consisting of a set X and a collection $\mathcal{R}$ of rules of inference over X. Given a set $Z \cup \{x\} \subseteq X$, x is said to **follow from** Z **by an application of a rule of inference** $R = \langle Y, y\rangle \in \mathcal{R}$, if $Y \subseteq Z$ and $x = y$. An $\mathcal{E}$**-proof of** x **from** Z is a sequence $x_0, x_1, \ldots, x_{n-1} \in X$, such that $x_{n-1} = x$ and, for all $i < n$, $x_i \in Z$ or x_i follows from $\{x_0, \ldots, x_{i-1}\}$ by an application of a rule of inference in $\mathcal{R}$. If there exists an $\mathcal{E}$-proof of x from Z, we write $Z \vdash_{\mathcal{E}} x$ and say that Z $\mathcal{E}$**-entails** x or that x is $\mathcal{E}$**-provable from** Z. The **entailment** associated with the entailment system $\mathcal{E}$ is the relation $\vdash_{\mathcal{E}} \subseteq \mathcal{P}(X) \times X$, defined by

$$Z \vdash_{\mathcal{E}} x \quad \text{iff} \quad \text{there exists an } \mathcal{E}\text{-proof of } x \text{ from } Z.$$

Finally, we write $Z^+ = C_{\mathcal{E}}(Z) := \{x \in X : Z \vdash_{\mathcal{E}} x\}$.

This definition of an entailment system encompasses four broad categories of important examples:

1) **Hierarchical Ontologies or Graphs:** A hierarchical ontology is a directed graph $G = \langle V, E\rangle$, where V is the set of nodes and E the set of directed edges, i.e., $E \subseteq V \times V$. When dealing with hierarchies, we are interested in discovering reachability relations between nodes based on the information stored in the directed edges of G. Therefore, the relevant entailment system in this case is $\mathcal{E} = \langle X, \mathcal{R}\rangle$, where $X = V \times V$ and $\mathcal{R}$ consists of the rule of inference (schema)
$$\frac{(x, y), (y, z)}{(x, z)}.$$
Clearly, the $\mathcal{E}$-closure of E is the transitive closure of the set E of directed edges of G.
Hierarchical ontologies cover a broad range of ontologies that are used in practice [17].
2) **Deductive Systems:** A **deductive system** $\mathcal{S} = \langle \mathcal{L}, \vdash_{\mathcal{S}}\rangle$ consists of a logical language type $\mathcal{L}$, i.e., a set of logical connectives $\mathcal{L}$ and a finitary and structural[2] consequence relation $\vdash_{\mathcal{S}} \subseteq \mathcal{P}(\mathrm{Fm}_{\mathcal{L}}(V)) \times \mathrm{Fm}_{\mathcal{L}}(V)$, where $\mathrm{Fm}_{\mathcal{L}}(V)$ denotes the collection of all formulas formed using propositional variables in a fixed denumerable set V and the logical connectives in the language $\mathcal{L}$. It is well-known that the entailment relation $\vdash_{\mathcal{S}}$ is induced by a collection $\mathcal{R}_{\mathcal{S}}$ of rules of inference over $\mathrm{Fm}_{\mathcal{L}}(V)$. Therefore, $\mathcal{S}$ may be represented by the entailment system $\mathcal{E}(\mathcal{S}) = \langle \mathrm{Fm}_{\mathcal{L}}(V), \mathcal{R}_{\mathcal{S}}\rangle$.
Deductive systems include propositional knowledge bases, which have many applications in AI and information systems [7].
3) **Hypergraphical Knowledge Bases:** This is a very general framework developed in [16] to deal with secrecy-preserving reasoning for ontologies expressed as hypergraphs. The key idea is to fix a set V of vertices and set $X = \mathcal{P}(V)$. Entailment is specified by a collection $\mathcal{R}$ of rules of inference over X. The resulting entailment system is $\mathcal{E} = \langle X, \mathcal{R}\rangle$. As a specific example of a rule of inference, consider the 1-*exclusive closure*, which is the property P ="for all $E, F, D \subseteq V$, such that $D \subseteq E \cap F$ and $|D| = 1$, $(E \cup F)\backslash D$ follows from E, F". This property may be expressed by the rule of inference (schema)
$$\frac{\{x_1, \ldots, x_n, z\}, \{z, y_1, \ldots, y_m\}}{\{x_1, \ldots, x_n, y_1, \ldots, y_m\}}.$$
Hypergraphical knowledge bases can be used to model connectivity among resources on the web [9].

2. The consequence relation $\vdash_{\mathcal{S}}$ is **structural** if $\Gamma \vdash_{\mathcal{S}} \phi$ implies $h(\Gamma) \vdash_{\mathcal{S}} h(\phi)$ for every substitution h.

4) **RDFS Knowledge Bases:** This very important entailment system will be discussed in detail in Section 6. It allows exploiting a sound and complete inference system for RDFS entailment to perform secrecy-preserving reasoning with arbitrary RDFS knowledge bases containing sensitive information. This framework makes it possible to handle a large number of "real-world" knowledge bases that are currently available in the semantic web.

Recall that in Section 2, we have defined the notions of a knowledge base, reasoner, secrecy set, secrecy envelopes and secrecy-preserving reasoners over an inference system $\mathcal{S}$. By replacing $\mathcal{S}$ by an arbitrary entailment system $\mathcal{E}$, we obtain, analogously, the same notions over $\mathcal{E}$.

We present an example of secrecy-preserving reasoning for a knowledge base $\mathbf{K} = \langle K, B, \mathrm{Fm}_{\mathcal{L}}(V), \{Y, U\}\rangle$ based on a deductive system $\mathcal{S} = \langle \mathcal{L}, \vdash_{\mathcal{S}}\rangle$. This is followed by an example of a secrecy-preserving reasoning with a hypergraphical knowledge base. An example of secrecy-preserving reasoning with the RDFS system, which forms one of the key applications of our framework because of its wide applicability in the existing semantic web, will be presented in Section 6.

Example 2: Suppose that $\mathcal{S}_1 = \langle \mathcal{L}, \vdash_{\mathcal{S}_1}\rangle, \mathcal{S}_2 = \langle \mathcal{L}, \vdash_{\mathcal{S}_2}\rangle$ are two deductive systems over the same language type $\mathcal{L}$, such that $\vdash_{\mathcal{S}_1} \subseteq \vdash_{\mathcal{S}_2}$, i.e., for all $\Gamma \cup \{\phi\} \subseteq \mathrm{Fm}_{\mathcal{L}}(V)$, $\Gamma \vdash_{\mathcal{S}_1} \phi$ implies $\Gamma \vdash_{\mathcal{S}_2} \phi$. In this case, there exist collections $\mathcal{R}_1, \mathcal{R}_2$ of rules of inference, with $\mathcal{R}_1 \subseteq \mathcal{R}_2$, such that $\mathcal{E}(\mathcal{S}_1) = \langle \mathrm{Fm}_{\mathcal{L}}(V), \mathcal{R}_1\rangle$ and $\mathcal{E}(\mathcal{S}_2) = \langle \mathrm{Fm}_{\mathcal{L}}(V), \mathcal{R}_2\rangle$. Let us further assume that there exists $\phi \in \mathrm{Fm}_{\mathcal{L}}(V)$, such that $\vdash_{\mathcal{S}_2} \phi$ but $\not\vdash_{\mathcal{S}_1} \phi$. Define

$$\mathrm{Thm}_{\mathcal{S}_1} = \{\phi \in \mathrm{Fm}_{\mathcal{L}}(V) : \vdash_{\mathcal{S}_1} \phi\}$$

and, similarly,

$$\mathrm{Thm}_{\mathcal{S}_2} = \{\phi \in \mathrm{Fm}_{\mathcal{L}}(V) : \vdash_{\mathcal{S}_2} \phi\}.$$

Consider the knowledge base

$$\mathbf{K} = \langle \mathrm{Thm}_{\mathcal{S}_2}, \mathrm{Thm}_{\mathcal{S}_1}, \mathrm{Fm}_{\mathcal{L}}(V), \{Y, U\}\rangle,$$

with secrecy set $S = \mathrm{Thm}_{\mathcal{S}_2} \backslash \mathrm{Thm}^{+}_{\mathcal{S}_1}$, the closure taken with respect to $\mathcal{S}_2$. According to the general framework developed here (and assuming, for simplicity, that $\mathcal{L}$ does not contain negation), a reasoner $R := R_{\mathcal{E}(\mathcal{S}_2)}(\mathbf{K})$ **for K over** $\mathcal{E}(\mathcal{S}_2)$ is a mapping $R : \mathrm{Fm}_{\mathcal{L}}(V) \to \{Y, U\}$, that satisfies (all closures referring to $\mathcal{S}_2$)

- Invariance: If $\langle \phi, \psi\rangle \in \Lambda(\mathcal{E}(\mathcal{S}_2))$, then $R(\phi) = R(\psi)$, for all $\phi, \psi \in \mathrm{Fm}_{\mathcal{L}}(V)$;
- Yes-Axiom: $\mathrm{Thm}^{+}_{\mathcal{S}_1} \subseteq \mathrm{Fm}_{\mathcal{L}}(V)_Y \subseteq \mathrm{Thm}_{\mathcal{S}_2}$.

In this case there is a unique secrecy envelope E_S for S, namely, the set $E_S = S = \mathrm{Thm}_{\mathcal{S}_2} \backslash \mathrm{Thm}^{+}_{\mathcal{S}_1}$, which can easily be seen to satisfy both the Enveloping and the Secrecy Axioms. Thus, the only secrecy-preserving reasoner for $\mathbf{K}$ and S is the function $R : \mathrm{Fm}_{\mathcal{L}}(V) \to \{Y, U\}$ defined by

$$R(\phi) = \begin{cases} Y, & \text{if } \phi \in \mathrm{Thm}^{+}_{\mathcal{S}_1} \\ U, & \text{if } \phi \in \mathrm{Fm}_{\mathcal{L}}(V) \backslash \mathrm{Thm}^{+}_{\mathcal{S}_1} \end{cases}.$$

□

We turn, now, to an example of secrecy-preserving reasoning for a hypergraphical ontology:

Example 3: Consider the hypergraph described pictorially in Figure 1. We describe it formally. Its set of vertices is $V = \{1, 2, 3, 4, 5, 6\}$. Let $X = \mathcal{P}(V)$ be the powerset of V. Consider the entailment system $\mathcal{E} = \langle X, \mathcal{R}\rangle$, where $\mathcal{R}$ consists of the inference rule schemas

$$\frac{\{x_1, \ldots, x_n, y\}, \{y, z_1, \ldots, z_m\}}{\{x_1, \ldots, x_n, z_1, \ldots, z_m\}},$$

for $1 \leq n, m \leq 5$. As was mentioned in Section 3, these rules are collectively referred to as 1-*exclusive closure*. Define $K = \{\{1, 2, 3\}, \{3, 4, 5\}, \{5, 6\}, \{1, 6\}\}$, $B = \emptyset$, $Q = X$ and $A = \{Y, U\}$. Let $\mathbf{K} = \langle K, B, Q, A\rangle$ be a knowledge base over $\mathcal{E}$. Assume that $S = \{\{1, 6\}\}$, which is depicted in Figure 1 by the dotted edge. It is easily seen that 1-exclusive closure allows the

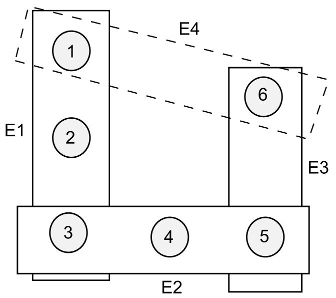

Figure 1. Illustration of a Hypergraph.

following two derivations:

$$\frac{\{1, 2, 3\}, \{3, 4, 5\}}{\{1, 2, 4, 5\}}, \qquad \frac{\{3, 4, 5\}, \{5, 6\}}{\{3, 4, 6\}}.$$

Suppose that we want to perform secrecy-preserving reachability reasoning, i.e., infer conclusions about whether a given vertex is reachable from another vertex by following a sequence of partially overlapping edges, without revealing the secret edge. Then, it is clear that a secrecy-preserving reasoner for $\mathbf{K}$ and S cannot answer "Y" to all three queries concerning the non-secret edges of K, since, in that case, it would compromise the secret information that $\{1, 6\}$ is an edge in the hypergraph. $\square$

The problem of computing the secrecy envelopes is addressed in the next section.

4. Order-Induced Secrecy-preserving Reasoners

4.1. The General Case

In this section, based on the general framework of secrecy-preserving reasoning with entailment systems, we develop an algorithm that produces secrecy-preserving reasoners in a wide variety of contexts. The algorithm has the advantage that it produces a minimal secrecy envelope for a given secrecy set, or, equivalently, it is maximally informative, in the sense that it responds with a minimal set of U-answers without risking disclosure of sensitive information.

Let $\mathcal{E} = \langle X, \mathcal{R} \rangle$ be an entailment system and $\mathbf{K} = \langle K, B, Q, A \rangle$ a knowledge base over $\mathcal{E}$, such that $Q = X$ and $A = \{Y, U\}$. Assume, moreover, that S is a secrecy set for $\mathbf{K}$. A secrecy-preserving reasoner $R : Q \to A$ for $\mathbf{K}$ and S is said to be **trivial** if $Q_Y = B^+$, i.e., if it answers Y only to the queries that are in or follow from the browsable part of $\mathbf{K}$. Clearly, such a reasoner is the "least informative" in the sense that it does not provide the querying agent any information in addition to the one that he can either browse or infer directly from browsable information. Note that a reasoner is trivial if and only if the corresponding secrecy envelope $E_S = K^+ \backslash B^+$, i.e., iff E_S is maximum (since, by definition, $S \subseteq E_S \subseteq K^+ \backslash B^+$). It is the goal of this section to show that, under mild assumptions on $\mathbf{K}$ and S, there always exist non-trivial secrecy-preserving reasoners for $\mathbf{K}$ and S and, moreover, to provide an algorithm that produces such a reasoner, if those conditions are satisfied.

The first assumption is that the set X of the entailment system $\mathcal{E} = \langle X, \mathcal{R} \rangle$ under consideration (which coincides with the set of queries Q of $\mathbf{K}$) is effectively enumerable, in which case the entailment system $\mathcal{E}$ is termed **effectively enumerable**. Let $\sigma = \langle x_0, x_1, \ldots \rangle$ be such an effective enumeration.

The main idea behind the algorithm is to exploit the fixed ordering σ of X to compute, for every $i > 0$, a finite set Q_Y^{i-1}, such that, in case $x_i \in K^+$, the browsable part B of the knowledge base together with Q_Y^{i-1} will determine whether the answer to x_i should be Y or U. Intuitively, Q_Y^{i-1} will be the collection of all x_j's, with $j < i$, to which the reasoner is supposed to answer Y. If $x_i \in K^+$ and $(B \cup Q_Y^{i-1} \cup \{x_i\})^+ \cap S = \emptyset$, then the answer to x_i is Y and it is U, otherwise. This method protects S because it avoids returning Y answers to queries, which, taken together with previously revealed information, risk the disclosure of secret knowledge.

The detailed algorithm is as follows:

`INPUT:` x_k;
$Q_Y^{-1} = \emptyset$;
For $i = 0$ to k do
  If ($x_i \in K^+$ and $(B \cup Q_Y^{i-1} \cup \{x_i\})^+ \cap S = \emptyset$)
    then $Q_Y^i = Q_Y^{i-1} \cup \{x_i\}$;
If ($x_k \in Q_Y^k$) then $R(x_k) = Y$ else $R(x_k) = U$;
Return $R(x_k)$;

Note that, according to the algorithm, the Y-answer set is $Q_Y = \bigcup_{i \geq 0} Q_Y^i$. In addition, the secrecy envelope associated with the reasoner induced by the algorithm is $E_S = K^+ \backslash Q_Y$.

In the following theorem, we show that, if there exists $x \in K^+ \backslash B^+$, such that $(B \cup \{x\})^+ \cap S = \emptyset$, then the function $R_\sigma = R : X \to A$, defined by the algorithm, is a non-trivial secrecy-preserving reasoner for $\mathbf{K}$ and S.

Theorem 2: Let $\mathbf{K} = \langle K, B, X, A \rangle$ be a knowledge base over an effectively enumerable entailment system $\mathcal{E} = \langle X, \mathcal{R} \rangle$ and S a given secrecy set for $\mathbf{K}$.

(i) Then $R_\sigma : X \to A$ is a secrecy-preserving reasoner for $\mathbf{K}$ and S.
(ii) If, in addition, there exists $x \in K^+ \backslash B^+$, such that $(B \cup \{x\})^+ \cap S = \emptyset$, then R_σ is non-trivial.

Proof:

(i) The first condition in the "If" statement of the algorithm ensures that $Q_Y \subseteq K^+$, i.e., that the Yes-Axiom for a reasoner is satisfied.
Suppose, for the sake of obtaining a contradiction, that R is not privacy-preserving for $\mathbf{K}$ and S. Since the Enveloping Axiom is obviously satisfied, this implies that $Q_Y^+ \cap S \neq \emptyset$. Thus, there exists $x_n \in S$, such that $x_n \in Q_Y^+$. Hence, for some $x_{i_0}, \ldots, x_{i_m} \in Q_Y$, $x_n \in \{x_{i_0}, \ldots, x_{i_m}\}^+$. Assume, without loss of generality, that $i_0 < i_1 < \cdots < i_m$. Then, $x_n \in (B \cup \{x_{i_0}, \ldots, x_{i_m}\})^+ \cap S \subseteq (B \cup Q_Y^{i_m})^+ \cap S$. Thus $(B \cup Q_Y^{i_m})^+ \cap S \neq \emptyset$, whence, by the algorithm, $R(x_{i_m}) = U$, which is a contradiction.
(ii) Let $k = \min\{i \in \mathbb{N} : x_i \in K^+ \backslash B^+$ and $(B \cup \{x_i\})^+ \cap S = \emptyset\}$. Then, $x_k \notin B^+$ and, by the

algorithm, we have $R(x_k) = Y$. Thus, R is non-trivial.

$\square$

Theorem 2 shows that, for each effective enumeration of the set X of elements of $\mathcal{E}$, one obtains a secrecy-preserving reasoner for $\mathbf{K}$ and S. A reasoner in this family of reasoners will be referred to as an **order-induced reasoner**.

Order-induced secrecy-preserving reasoners have the important property that they capture exactly "maximal informativeness". More precisely, as the following results show, every order-induced secrecy-preserving reasoner is maximal in the sense that it answers Y to a largest possible subset of X without revealing secret information. Furthermore, every maximal reasoner in this sense is order-induced.

Definition 3: Let $\mathbf{K} = \langle K, B, X, A\rangle$ be a knowledge base over an entailment system $\mathcal{E} = \langle X, \mathcal{R}\rangle$ and S a secrecy set for $\mathbf{K}$. A secrecy-preserving reasoner $R : X \to A$ for $\mathbf{K}$ and S is called **maximal** if Q_Y is maximal among the family of all subsets $Z \subseteq X$, that satisfy:

- $B^+ \subseteq Z \subseteq K^+$ and
- $Z^+ \cap S = \emptyset$.

The following two theorems clarify the connection between order-induced secrecy-preserving reasoners and maximal secrecy-preserving reasoners.

Theorem 4: Let $\mathbf{K} = \langle K, B, X, A\rangle$ be a knowledge base over an effectively enumerable entailment system $\mathcal{E} = \langle X, \mathcal{R}\rangle$ and S a secrecy set for $\mathbf{K}$. If $R : X \to \{Y, U\}$ is an order-induced secrecy-preserving reasoner for $\mathbf{K}$ and S, then R is maximal.

Proof: Suppose that R is an order-induced privacy-preserving reasoner for $\mathbf{K}$ over $\mathcal{E}$ and let $x_0, x_1, \ldots$ be the ordering of X inducing the reasoner R. For the sake of obtaining a contradiction, suppose that R is not maximal. Thus, there exists $x_i \in X\backslash Q_Y$, such that $x_i \in K^+$ and $(Q_Y \cup \{x_i\})^+ \cap S = \emptyset$. Let n be minimum among all such i. Then, in the notation of the algorithm, we have $x_n \in K^+$ and $(Q_Y^{n-1} \cup \{x_n\})^+ \cap S = \emptyset$. This implies that $R(x_n) = Y$, contradicting the fact that $x_n \in X\backslash Q_Y$. $\square$

A somewhat more surprising fact is that every maximal secrecy-preserving reasoner for $\mathbf{K}$ and S is order-induced. The proof of Theorem 5 is more involved and will be omitted.

Theorem 5: Let $\mathbf{K} = \langle K, B, X, A\rangle$ be a knowledge base over an effectively enumerable entailment system $\mathcal{E} = \langle X, \mathcal{R}\rangle$ and S a secrecy set for $\mathbf{K}$. Every maximal secrecy-preserving reasoner $R : X \to \{Y, U\}$ for $\mathbf{K}$ and S is order-induced.

Next, we seek to formalize a measure for comparing various order-induced secrecy-preserving reasoners with respect to their "informativeness".

Definition 6: Let $\mathbf{K} = \langle K, B, X, A\rangle$ be a knowledge base over an effectively enumerable entailment system $\mathcal{E} = \langle X, \mathcal{R}\rangle$ and S a secrecy set for $\mathbf{K}$. Let σ and τ be two orderings of X. We say that σ is **less informative than** τ or that τ is **more informative than** σ, written $\sigma \preccurlyeq \tau$, if

$$R_\sigma(x) = Y \quad \text{implies} \quad R_\tau(x) = Y, \quad \text{for all } x \in X.$$

Theorem 4 has the following corollary:

Corollary 7: Let $\mathbf{K} = \langle K, B, X, A\rangle$ be a knowledge base over an effectively enumerable entailment system $\mathcal{E} = \langle X, \mathcal{R}\rangle$ and S a secrecy set for $\mathbf{K}$. For all orderings σ, τ of X, we have either $R_\sigma = R_\tau$ or σ and τ are incomparable in the $\preccurlyeq$-ordering.

The following examples show that both cases listed in Corollary 7, concerning order-induced secrecy-preserving reasoners for various $\mathbf{K}$ and S, may actually occur.

Example 4: Let $\mathcal{E} = \langle X, \mathcal{R}\rangle$ be the entailment system defined as follows:

$$X := \{x_0, x_1, y_0, y_1, z_0, z_1, w_0, w_1\}.$$

Consider the following linear ordering $<$ of the elements of X: $y_0, z_0, y_1, z_1, x_0, w_0, x_1, w_1$. $\mathcal{R}$ is any set of inference rules on X that induces the closure operator defined, for all $Y \subseteq X$, by

$$Y^+ = \{u \in X : (\exists v \in Y)(v \leq u)\}.$$

For instance, we may take $\mathcal{R}$ to consist of all rules of the form $\frac{Y}{y}$, with $y \in Y^+$. Furthermore, consider the knowledge base $\mathbf{K} = \langle K, B, X, A\rangle$ over $\mathcal{E}$ defined by $K = X$ and $B = \{x_0, x_1\}$. Let $S = \{y_0, y_1\}$ be a secrecy set for $\mathbf{K}$. Since

$$\begin{aligned} B^+ \cap S &= \{x_0, x_1\}^+ \cap \{y_0, y_1\} \\ &= \{x_0, x_1, w_0, w_1\} \cap \{y_0, y_1\} \\ &= \emptyset, \end{aligned}$$

there exists a secrecy-preserving reasoner for $\mathbf{K}$ and S. We show that every ordering of the elements of X yields the same order-induced secrecy-preserving reasoner.

First, note that the reasoner induced by the standard ordering $<$, defined above, would give $Q_U = \{y_0, y_1, z_0\}$ and $Q_Y = \{z_1, x_0, w_0, x_1, w_1\}$. Note also that any other secrecy-preserving reasoner must answer

U to all three of y_0, y_1 and z_0: to the first two because they are in S and to the third because it reveals the secret element y_1. Thus, the claim follows by Theorem 4, since any difference in the remaining answers of an order-induced secrecy-preserving reasoner would mean that the reasoner is non-maximal.

Finally, note that Theorem 5 implies that the reasoner R is the only maximal secrecy-preserving reasoner for $\mathbf{K}$ and S over $\mathcal{E}$. $\square$

The next example describes a knowledge base which possesses more than one order-induced secrecy-preserving reasoner.

Example 5: Consider the entailment system $\mathcal{E} = \langle X, \mathcal{R}\rangle$, defined by $X = \{p, q, r\}$ and $\mathcal{R}$ a set of rules of inference inducing the closure operator given by

$$\emptyset^+ = \emptyset, \{p\}^+ = \{p\}, \{q\}^+ = \{q\}, \{r\}^+ = \{r\},$$

$$Y^+ = \{p, q, r\}, \text{ all other } Y \subseteq X.$$

Let $\mathbf{K} = \langle X, \emptyset, X, A\rangle$ and consider the secrecy set $S = \{r\}$ in $\mathbf{K}$. We have $B^+ \cap S = \emptyset$ and, therefore, there exist secrecy-preserving reasoners for $\mathbf{K}$ and S. Consider now the two orderings

$$p < q < r \qquad q \lessdot p \lessdot r.$$

Then, the $<$-order-induced secrecy-preserving reasoner for $\mathbf{K}$ and S has $Q_Y^< = \{p\}$ and $Q_U^< = \{q, r\}$, whereas the $\lessdot$-order-induced secrecy-preserving reasoner for $\mathbf{K}$ and S yields $Q_Y^\lessdot = \{q\}$ and $Q_U^\lessdot = \{p, r\}$.

Note that, if the knowledge base is changed so that $B = \{p\}$, then the fact that $B^+ \subseteq Q_Y$ would force both order-induced secrecy-preserving reasoners to answer Y the query p and U the query q. Thus, in that case, the two reasoners would be identical. $\square$

4.2. Hierarchical Knowledge Bases

Suppose that $\mathcal{E} = \langle X, \mathcal{R}\rangle$ is an effectively enumerable entailment system. Let $\mathbf{K} = \langle K, B, X, A\rangle$ be a knowledge base over $\mathcal{E}$ and S a secrecy set for $\mathbf{K}$. By the results of the previous section, we know that, in general, an order-induced secrecy-preserving reasoner for $\mathbf{K}$ and S is maximal and that two different total orderings of X give rise to potentially different maximal secrecy-preserving reasoners for $\mathbf{K}$ over $\mathcal{S}$. In this section we show that this remains the case even when one restricts attention to hierarchical knowledge bases. These are knowledge bases that consist of a directed graph $G = \langle V, E\rangle$, some of whose edges are considered browsable and some secret. We define an entailment system whose rules induce the reachability relation over V. Specifically, let $\mathcal{E} = \langle V \times V, \mathcal{R}\rangle$, where $\mathcal{R}$ consists of all rules of inference of the form

$$\frac{}{(x, x)} \quad \text{and} \quad \frac{(x, y), (y, z)}{(x, z)}. \tag{2}$$

Thus, given a collection $X \subseteq V \times V$ and a pair $(y, z) \in V \times V$, $(y, z) \in X^+$ if and only if $y = z$ or there exists a sequence of pairs $(w_0, w_1), (w_1, w_2), \ldots, (w_{n-1}, w_n) \in X$, such that $w_0 = y$ and $w_n = z$.

A **hierarchical knowledge base** or **hierarchical ontology** over $\mathcal{E}$ is a knowledge base

$$\mathbf{K} = \langle E, B, V \times V, \{Y, U\}\rangle,$$

where E is the set of edges of G and $B \subseteq E$. Let $S \subseteq E^+$ be a secrecy set for $\mathbf{K}$.

By Theorem 4, we know that at least one maximal secrecy-preserving reasoner for $\mathbf{K}$ and S exists and may be obtained by considering the order-induced secrecy-preserving reasoner associated with an ordering of $V \times V$. In view of Examples 4 and 5, we want to address the question of uniqueness in the special case of hierarchical ontologies.

We show in the next example that there exists a hierarchical knowledge base $\mathbf{K}$ over $\mathcal{E}$, a secrecy set S for $\mathbf{K}$ and two linear orderings $<$ and $\lessdot$ on $V \times V$, such that the two maximal order-induced secrecy-preserving reasoners for $\mathbf{K}$ over $\mathcal{E}$ are incomparable. Hence, even in the relatively simple framework of hierarchical ontologies, both outcomes expected by the conclusion of Corollary 7 actually occur.

Example 6: Consider the vertex set $V = \{x, y, z, w, u\}$ and the entailment system $\mathcal{E} = \langle V \times V, \mathcal{R}\rangle$ defined by Rules (2). Let $\mathbf{K} = \langle E, B, V \times V, A\rangle$ be the knowledge base over $\mathcal{E}$ defined by $B = \{(x, y), (z, w)\}$ and consider the secrecy set $S = \{(y, z), (w, u), (x, u)\}$ for $\mathbf{K}$. This knowledge base is illustrated in Figure 2, where the edges in B are solid and the edges in S are dashed. Consider the

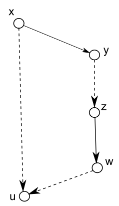

Figure 2. The hierarchical knowledge base E of Example 6.

linear ordering of the vertices x, y, z, w, u and let $<$ denote the lexicographic ordering on $V \times V$ and $\lessdot$

the reverse lexicographic ordering on $V \times V$. Then it is not difficult to see by application of the order-induced secrecy-preserving reasoning algorithm that the $<$-order-induced and the $\lessdot$-order-induced secrecy-preserving reasoners $R^{<}$ and $R^{\lessdot}$ for $\mathbf{K}$ and S are given, respectively, by

$R^{<}$	x	y	z	w	u
x	Y	Y	Y	Y	U
y	U	Y	U	Y	U
z	U	U	Y	Y	U
w	U	U	U	Y	U
u	U	U	U	U	Y

$R^{\lessdot}$	x	y	z	w	u
x	Y	Y	U	Y	U
y	U	Y	U	Y	U
z	U	U	Y	Y	Y
w	U	U	U	Y	U
u	U	U	U	U	Y

The Y answers for each of the two reasoners reveal the information about the edges in E (not showing loops) that is illustrated in Figure 3. □

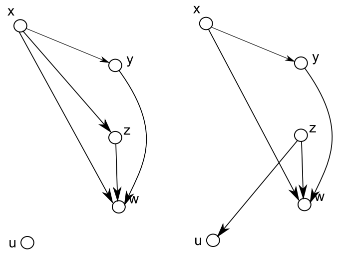

Figure 3. The information revealed by the secrecy-preserving reasoners $R^{<}$ and $R^{\lessdot}$, respectively.

5. Secrecy-preserving Mappings

Let $\mathcal{E}_1 = \langle X_1, \mathcal{R}_1 \rangle$ and $\mathcal{E}_2 = \langle X_2, \mathcal{R}_2 \rangle$ be two entailment systems, $\mathbf{K}_1 = \langle K_1, B_1, X_1, A \rangle$ and $\mathbf{K}_2 = \langle K_2, B_2, X_2, A \rangle$ knowledge bases over $\mathcal{E}_1$ and $\mathcal{E}_2$, respectively, and S_1, S_2 secrecy sets for $\mathbf{K}_1, \mathbf{K}_2$, respectively. Henceforth, the notation $(\mathbf{K}, S)$ will be adopted for the pair consisting of a knowledge base $\mathbf{K}$ and a secrecy set S for $\mathbf{K}$. Informally speaking, a secrecy-preserving map from $(\mathbf{K}_1, S_1)$ to $(\mathbf{K}_2, S_2)$ should be a function that would help construct a secrecy-preserving reasoner for $\mathbf{K}_2$ and S_2 whenever a secrecy-preserving reasoner for $\mathbf{K}_1$ and S_1 is available. These functions are very useful because they provide a way to "reuse" reasoners. Suppose, for instance, that one has available some "off-the-shelf" reasoner for a knowledge base $\mathbf{K}$ with secrecy set S. A user who wants to perform secrecy-preserving reasoning in a different application context, say over a knowledge base $\mathbf{K}'$ with secrecy set S', would only have to construct a secrecy-preserving mapping from $(\mathbf{K}, S)$ to $(\mathbf{K}', S')$. In this section, we introduce such secrecy-preserving mappings and prove some interesting results-mainly sufficient conditions-about them.

Suppose that $\mathcal{E}_1 = \langle X_1, \mathcal{R}_1 \rangle$, $\mathcal{E}_2 = \langle X_2, \mathcal{R}_2 \rangle$ are two entailment systems and $\mathbf{K}_1 = \langle K_1, B_1, Q_1, A \rangle, \mathbf{K}_2 = \langle K_2, B_2, Q_2, A \rangle$ are two knowledge bases over $\mathcal{E}_1, \mathcal{E}_2$, respectively, where, for simplicity, we assume that $Q_1 = X_1, Q_2 = X_2$ and $A = \{Y, U\}$. Let, also, $S_1 \subseteq K_1^+$ and $S_2 \subseteq K_2^+$ be two secrecy sets for $\mathbf{K}_1$ and $\mathbf{K}_2$, respectively.

Definition 8: A **secrecy-preserving mapping** $h : (\mathbf{K}_1, S_1) \to (\mathbf{K}_2, S_2)$ is a function $h : X_2 \to X_1$, such that, for every secrecy-preserving reasoner $R_{\mathcal{E}_1}(\mathbf{K}_1) : X_1 \to A$ for $\mathbf{K}_1$ and S_1, the induced function $R_{\mathcal{E}_2}(\mathbf{K}_2) := R_{\mathcal{E}_1}(\mathbf{K}_1) \circ h : X_2 \to A$ is a secrecy-preserving reasoner for $\mathbf{K}_2$ and S_2.

Pictorially, Definition 8 requires commutativity of the following diagram, such that, whenever $R_{\mathcal{E}_1}(\mathbf{K}_1)$ is secrecy-preserving, $R_{\mathcal{E}_2}(\mathbf{K}_2)$ is also secrecy-preserving.

$$\begin{tikzcd} X_1 \arrow[dr, "R_{\mathcal{E}_1}(\mathbf{K}_1)"'] & & X_2 \arrow[ll, "h"'] \arrow[dl, "R_{\mathcal{E}_2}(\mathbf{K}_2)"] \\ & A & \end{tikzcd}$$

In the remainder of the section, we let $\mathcal{E}_1 = \langle X_1, \mathcal{R}_1 \rangle$ and $\mathcal{E}_2 = \langle X_2, \mathcal{R}_2 \rangle$ be two entailment systems, $\mathbf{K}_1 = \langle K_1, B_1, X_1, A \rangle$, $\mathbf{K}_2 = \langle K_2, B_2, X_2, A \rangle$ two knowledge bases over $\mathcal{E}_1, \mathcal{E}_2$, respectively, and $h : X_2 \to X_1$ a mapping.

The following lemma provides a relationship between the Y-answer sets Q_Y^1 and Q_Y^2 of a secrecy-preserving reasoner $R_{\mathcal{E}_1}(\mathbf{K}_1)$ for $\mathbf{K}_1$ and S_1 and the induced reasoner $R_{\mathcal{E}_2}(\mathbf{K}_2) := R_{\mathcal{E}_1}(\mathbf{K}_1) \circ h$ for $\mathbf{K}_2$ and S_2.

Lemma 9: If $R_{\mathcal{E}_1}(\mathbf{K}_1) : X_1 \to A$ is a reasoner for $\mathbf{K}_1$ and $R_{\mathcal{E}_2}(\mathbf{K}_2) = R_{\mathcal{E}_1}(\mathbf{K}_1) \circ h$, with respective Y-query sets Q_Y^1, Q_Y^2, then $Q_Y^2 = h^{-1}(Q_Y^1)$.

Proof: Indeed we have, for all $x \in X_2$,

$$\begin{array}{lll} x \in Q_Y^2 & \text{iff} & R_{\mathcal{E}_2}(\mathbf{K}_2)(x) = Y \\ & \text{iff} & R_{\mathcal{E}_1}(\mathbf{K}_1)(h(x)) = Y \\ & \text{iff} & h(x) \in Q_Y^1 \\ & \text{iff} & x \in h^{-1}(Q_Y^1). \end{array}$$

□

The following proposition lists sufficient conditions for $R_{\mathcal{E}_1}(\mathbf{K}_1) \circ h$ to be a reasoner for $\mathbf{K}_2$, given a reasoner $R_{\mathcal{E}_1}(\mathbf{K}_1)$ for $\mathbf{K}_1$.

Proposition 10: Let $R_{\mathcal{E}_1}(\mathbf{K}_1) : X_1 \to A$ a reasoner for $\mathbf{K}_1$. Then $R_{\mathcal{E}_2}(\mathbf{K}_2) := R_{\mathcal{E}_1}(\mathbf{K}_1) \circ h : X_2 \to A$ is a reasoner for $\mathbf{K}_2$ if

1) For every $Y \cup \{y\} \subseteq X_2$, $y \in Y^+$ implies $h(y) \in h(Y)^+$;[3]
2) $B_2^+ \subseteq h^{-1}(Q_Y^1) \subseteq K_2^+$.

Proof: Suppose that $\langle y_1, y_2 \rangle \in \Lambda(\mathcal{E}_2)$. Then, $\{y_1\}^+ = \{y_2\}^+$ and, hence, by the hypothesis, $\{h(y_1)\}^+ = \{h(y_2)\}^+$. This shows that $\langle h(y_1), h(y_2) \rangle \in \Lambda(\mathcal{E}_1)$. Since $R_{\mathcal{E}_1}(\mathbf{K})$ is a reasoner, we obtain $R_{\mathcal{E}_1}(\mathbf{K}_1)(h(y_1)) = R_{\mathcal{E}_1}(\mathbf{K}_1)(h(y_2))$, showing that $R_{\mathcal{E}_2}(\mathbf{K}_2)(y_1) = R_{\mathcal{E}_2}(\mathbf{K}_2)(y_2)$. Hence $R_{\mathcal{E}_2}(\mathbf{K}_2)$ is invariant. The second condition, together with Lemma 9, implies that $B_2^+ \subseteq Q_Y^2 \subseteq K_2^+$. Thus, $R_{\mathcal{E}_2}(\mathbf{K}_2)$ is a valid reasoner for $\mathbf{K}_2$. □

Finally, in the next proposition we obtain sufficient conditions for a mapping $h : X_2 \to X_1$ to be a secrecy-preserving mapping.

Proposition 11: The mapping h is a secrecy-preserving mapping $h : (\mathbf{K}_1, S_1) \to (\mathbf{K}_2, S_2)$ if, for every $Y \cup \{y\} \subseteq X_2$, $y \in Y^+$ implies $h(y) \in h(Y)^+$, and, for every secrecy-preserving reasoner $R_{\mathcal{E}_1}(\mathbf{K}_1)$ for $\mathbf{K}_1$ and S_1,

1) $B_2^+ \subseteq h^{-1}(Q_Y^1) \subseteq K_2^+$;
2) $S_2 \subseteq h^{-1}(Q_U^1) \subseteq K_2^+ \backslash B_2^+$;
3) $(K_2^+ \backslash h^{-1}(Q_U^1))^+ \cap S_2 = \emptyset$.

Proof: Assume that $R_{\mathcal{E}_1}(\mathbf{K}_1) : X_1 \to A$ is a secrecy-preserving reasoner for $\mathbf{K}_1$ and S_1. Then, the first two assumptions combined with Proposition 10 show that $R_{\mathcal{E}_2}(\mathbf{K}_2) := R_{\mathcal{E}_1}(\mathbf{K}_1) \circ h : X_2 \to A$ is a reasoner for $\mathbf{K}_2$. The last two hypotheses show that it is a secrecy-preserving reasoner for $\mathbf{K}_2$ and S_2. Since this holds for every secrecy-preserving reasoner $R_{\mathcal{E}_1}(\mathbf{K}_1)$ for $\mathbf{K}_1$ and S_1, we conclude that $h : (\mathbf{K}_1, S_1) \to (\mathbf{K}_2, S_2)$ is a secrecy-preserving mapping. □

6. RDFS System

The basic syntactic components of RDF are the three disjoint infinite sets $\mathbf{U}, \mathbf{B}$ and $\mathbf{L}$ of **URI references**, **blank nodes** and **literals**, respectively. We follow [14] in this section in denoting unions of these sets by concatenating their names. An **RDF triple** is a triple $(s, p, o) \in \mathbf{UBL} \times \mathbf{U} \times \mathbf{UBL}$, where s is the **subject**, p the **predicate** and o the **object** of the RDF triple. An **RDF graph** is a set of RDF triples. The **universe** of an RDF graph G is the set of elements in $\mathbf{UBL}$ that occur in the triples of G and it is denoted by $\mathrm{univ}(G)$. The **vocabulary** of G is the set $\mathrm{voc}(G) = \mathrm{univ}(G) \cap \mathbf{UL}$. An RDF graph is **ground** if $\mathrm{univ}(G) = \mathrm{voc}(G)$, i.e., if it has no blank nodes.

3. The consequence operator $^+$ is context sensitive, always referring to the entailment system of which its argument is a subset.

Example 7: In Figure 4 we show part of an RDF ontology that describes the relations between authors, articles they authored and journals in which the articles were published. Friendships between authors are also

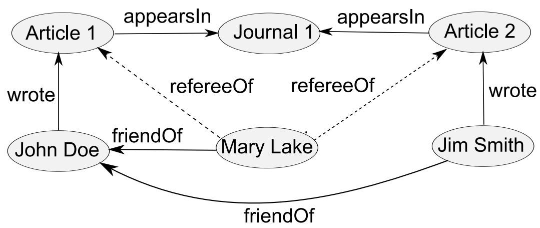

Figure 4. RDF Knowledge Base of Example 7.

documented as well as refereeing of given articles by specific authors. Such a knowledge base may be used, for instance, to decide on possible conflicts of interest when assigning referees to articles or, alternatively, may be part of a scientific-social network. The dashed lines represent information that the knowledge base administrator may want to keep secret, since refereeing is supposed to be a confidential process. The reader, we hope, would be able to imagine many other similar applications, where either because of confidentiality or due to privacy, security or copyright issues, various pieces of information in a given knowledge base may need to be kept secret. □

An **interpretation** over a vocabulary V is a septuple

$$\mathcal{I} = \langle \mathrm{Res}, \mathrm{Prop}, \mathrm{Class}, \mathrm{Ext}, \mathrm{CExt}, \mathrm{Lit}, \mathrm{Int} \rangle,$$

such that

1) Res is a nonempty set of **resources** called the domain or **universe** of $\mathcal{I}$;
2) Prop is a set of property names;
3) $\mathrm{Class} \subseteq \mathrm{Res}$ is a distinguished subset of Res identifying those resources that are classes of resources.
4) $\mathrm{Ext} : \mathrm{Prop} \to 2^{\mathrm{Res} \times \mathrm{Res}}$ is a mapping that assigns an extension to each property name;
5) $\mathrm{CExt} : \mathrm{Class} \to 2^{\mathrm{Res}}$ is a mapping that assigns a set of resources to each resource denoting a class;
6) $\mathrm{Lit} \subseteq \mathrm{Res}$ is the set of literal values, that contains all plain literals in $\mathbf{L} \cap V$;
7) $\mathrm{Int} : \mathbf{UL} \cap V \to \mathrm{Res} \cup \mathrm{Prop}$ is the **interpretation mapping**, that assigns a resource or property name to each element of $\mathbf{UL} \cap V$ with the restriction that Int is the identity on plain literals and assigns an element in Res to elements in $\mathbf{L}$.

According to [13], [10] (see also [14]), a ground triple (s, p, o) in G is **true under** $\mathcal{I}$ if p is interpreted as a property name, s and o are interpreted as resources and the interpretation of (s, o) belongs to the extension of p. When dealing with arbitrary (non-ground) RDF triples, e.g., (X, p, o), with $X \in \mathbf{B}$, is **true under** $\mathcal{I}$ if there exists a resource s, such that (s, p, o) is true under $\mathcal{I}$. For the RDF graph, each blank node must be interpreted as the same resource wherever it appears throughout G. Now RDF entailment is defined as usual based on the satisfaction of an RDF graph under certain interpretations. Namely, those interpretations that model appropriately the RDF designated vocabulary and satisfy the set of RDF axiomatic triples.

A simple fragment of RDFS, called ρdf fragment, was introduced and studied in some detail in [14]. We use it, for simplicity to illustrate how out secrecy-preserving framework of Section 3 can be used to reason with an RDFS knowledge base containing secret information. However, the reader should notice that our technique could be applied seamlessly to reasoning with the entire RDFS system [13], not merely with this small fragment of its vocabulary.

The ρ**df vocabulary** is defined by

$$\rho\mathrm{df} = \{\mathtt{sp}, \mathtt{sc}, \mathtt{type}, \mathtt{dom}, \mathtt{range}\}.$$

An RDF graph G over ρdf will be termed a ρ**df graph** or, simply a **graph**. An interpretation $\mathcal{I}$ is a **model** of G, denoted by $\mathcal{I} \models_{\rho\mathrm{df}} G$ if it is an interpretation over $\rho\mathrm{df} \cup \mathrm{univ}(G)$ that satisfies various conditions (see [14]). For example, the conditions pertaining to the keyword sp (subproperty) say that $\mathrm{Ext}(\mathrm{Int}(\mathtt{sp}))$ is transitive and reflexive over Prop and that, if $(x, y) \in \mathrm{Ext}(\mathrm{Int}(\mathtt{sp}))$, then $x, y \in \mathrm{Prop}$ and $\mathrm{Ext}(x) \subseteq \mathrm{Ext}(y)$.

In [14] (see also [15], [13]) a sound and complete deductive system for the ρdf fragment of RDFS was presented. Again, for the sake of providing an insight into the flavor of this system, let us mention that the two inference rules that specifically handle sp are

$$\frac{(\mathcal{A}, \mathtt{sp}, \mathcal{B})\ (\mathcal{B}, \mathtt{sp}, \mathcal{C})}{(\mathcal{A}, \mathtt{sp}, \mathcal{C})} \quad \frac{(\mathcal{A}, \mathtt{sp}, \mathcal{B})\ (\mathcal{X}, \mathcal{A}, \mathcal{Y})}{(\mathcal{X}, \mathcal{B}, \mathcal{Y})}.$$

Note, also, that there exists a similar sound and complete entailment system for the entire RDFS vocabulary. This was presented in [10] and later completed in [13]. We denote the ρdf entailment system of [14] by $\mathcal{E}_{\rho\mathrm{df}}$ and the corresponding RDFS entailment system of [10] by $\mathcal{E}_{\mathrm{RDFS}}$.

Let G be a ρdf graph and $B \subseteq G$. Let, also, $Q = T_{\rho\mathrm{df}}$, the collection of all ρdf terms and $A = \{Y, U\}$. Then, the quadruple $\mathbf{G} = \langle G, B, T_{\rho\mathrm{df}}, A\rangle$ will be referred to as a ρdf-**knowledge base**. In addition, consider $S \subseteq G^+$, a subset of the inferential closure of the ρdf graph G under $\mathcal{E}_{\rho\mathrm{df}}$. According to Section 3, a reasoner $R_{\rho\mathrm{df}}(\mathbf{G})$ for $\mathbf{G}$ over $\mathcal{E}_{\rho\mathrm{df}}$ is a mapping $R_{\rho\mathrm{df}}(\mathbf{G}) : T_{\rho\mathrm{df}} \to A$, that satisfies

- Invariance: If $\langle s, t\rangle \in \Lambda(\mathcal{E}_{\rho\mathrm{df}})$, then $R(s) = R(t)$, for all triples $s, t \in T_{\rho\mathrm{df}}$;
- Yes-Axiom: $B^+ \subseteq Q_Y \subseteq G^+$;

A secrecy envelope for the secrecy set of triples $S \subseteq G^+$ is a set of triples E_S, satisfying

- Enveloping Axiom: $S \subseteq E_S \subseteq G^+ \backslash B^+$;
- Secrecy Axiom: $(G^+ \backslash E_S)^+ \cap S = \emptyset$.

Then, the secrecy-preserving reasoner based on E_S is the function $R := R_{E_S} : T_{\rho\mathrm{df}} \to A$ defined, for all $t \in T_{\rho\mathrm{df}}$, by

$$R(t) = \begin{cases} Y, & \text{if } t \in G^+ \backslash E_S \\ U, & \text{otherwise} \end{cases}. \tag{3}$$

The fact that the ρdf entailment system is sound and complete with respect to the ρdf semantics ensures that, by using the entailment system $\mathcal{E}_{\rho\mathrm{df}}$, we capture exactly the semantic entailment of the ρdf fragment of RDF, as was intended in the original RDF specification.

The generalization of this process to RDFS entailment using the entailment system $\mathcal{E}_{\mathrm{RDFS}}$ is powerful enough for many of the knowledge bases that exist in the current semantic web.

7. Summary

In this paper, we introduced a *very general framework for performing secrecy-preserving reasoning* with knowledge bases containing secret information. To formally express arbitrary knowledge bases, we introduced the notion of an *entailment system*. Its generality allows us to capture many of the important examples of existing knowledge bases in the semantic web. These include hierarchical knowledge bases, where information may be represented in the form of a directed acyclic graph, knowledge bases expressible in some propositional language, e.g., classical propositional logic, hypergraphical knowledge bases, where information is representable in the form of a hypergraph, as well as knowledge bases expressed in some description logic or using the RDF paradigm. We showed that the advantage of our framework lies in being general enough to capture virtually all special cases of current interest, while, at the same time, scaling down to each of them in a simple and understandable way. We also presented an *algorithm for devising secrecy-preserving reasoners*, given a knowledge base and a secrecy set for it. The reasoners produced by the algorithm have the property that they hide the minimal possible amount of information without jeopardizing the sensitive knowledge. To provide a means to reuse secrecy-preserving reasoners across different knowledge bases, we introduce the notion of a *secrecy-preserving mapping* and formulate sufficient conditions ensuring its correct functionality. Finally, we place some emphasis in the way our framework can be used to perform secrecy-preserving reasoning with RDFS knowledge bases, since these comprise many of the existing knowledge bases on the semantic web. We are currently in the process of implementing and testing the efficiency of hierarchical and propositional secrecy-preserving reasoners. In the future, we are planning to use a DL reasoner to test secrecy-preserving reasoning in applications involving more powerful languages.

References

[1] Jie Bao, Giora Slutzki, and Vasant Honavar. Privacy-preserving reasoning on the semantic web. In *Web Intelligence*, pages 791–797, 2007.

[2] Elisa Bertino, Latifur R. Khan, Ravi S. Sandhu, and Bhavani M. Thuraisingham. Secure knowledge management: confidentiality, trust, and privacy. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 36(3):429–438, 2006.

[3] Piero A. Bonatti, Claudiu Duma, Norbert Fuchs, Wolfgang Nejdl, Daniel Olmedilla, Joachim Peer, and Nahid Shahmehri. Semantic web policies - a discussion of requirements and research issues. In *ESWC*, pages 712–724, 2006.

[4] Bernardo Cuenca Grau and Ian Horrocks. Privacy-preserving query answering in logic-based information systems. In *Proc. of the 18th Eur. Conf. on Artificial Intelligence (ECAI 2008)*, 2008.

[5] Csilla Farkas, Alexander Brodsky, and Sushil Jajodia. Unauthorized inferences in semi-structured databases. *Information Sciences*, 176(22):3269–3299, Nov. 2006.

[6] Josep Maria Font, Ramon Jansana, and Don Pigozzi. A survey of abstract algebraic logic. *Studia Logica*, 74(1-2):13–97, 2003.

[7] Michael R. Genesereth and Nils J. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers, Burlington, MA, 1988.

[8] Mark Giereth. On partial encryption of rdf-graphs. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2005.

[9] Jonathan Hayes and Claudio Gutiérrez. Bipartite graphs as intermediate model for rdf. In *International Semantic Web Conference*, pages 47–61, 2004.

[10] Patrick Hayes. Rdf semantics. Technical Report REC-rdf-mt-20040210, W3C, Cambridge, MA, 2004.

[11] Amit Jain and Csilla Farkas. Secure resource description framework: an access control model. In *SACMAT*, pages 121–129, 2006.

[12] Vladimir Kolovski, James A. Hendler, and Bijan Parsia. Analyzing web access control policies. In *WWW*, pages 677–686, 2007.

[13] Draltan Marin. A formalization of rdf (applications de la logique à la sémantique du web). http://www.dcc.uchile.cl/cgutierr/ftp/draltan.pdf, 2006.

[14] Sergio Muñoz, Jorge Pérez, and Claudio Gutiérrez. Minimal deductive systems for rdf. In *ESWC*, pages 53–67, 2007.

[15] Herman J. ter Horst. Completeness, decidability and complexity of entailment for rdf schema and a semantic extension involving the owl vocabulary. *J. Web Sem.*, 3(2-3):79–115, 2005.

[16] George Voutsadakis, Jie Bao, Giora Slutzki, and Vasant Honavar. Privacy-preserving reasoning for hypergraphical knowledge bases. *Technical Report*, 2008.

[17] Taowei David Wang, Bijan Parsia, and James A. Hendler. A survey of the web ontology landscape. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 682–694. Springer, 2006.