A mechanism for ontology confidentiality

P. A. Bonatti, I. M. Petrova and L. Sauro

Dept. of Electrical Engineering and Information Technologies
Università di Napoli “Federico II”

Abstract. We illustrate several novel attacks to the confidentiality of knowledge bases (KB). Then we introduce a new confidentiality model, sensitive enough to detect those attacks, and a method for constructing secure KB views.We identify safe approximations of the background knowledge exploited in the attacks; they can be used to reduce the complexity of constructing secure KB views. Finally we describe a prototype implementation of the new approach that suggests its applicability in practice.

1 Introduction

Ontology languages and Linked Open Data are increasingly being used to encode the private knowledge of companies and public organizations. Semantic Web techniques make possible to merge different sources of knowledge and extract implicit information, putting on risk security and privacy of individuals. Even the authors of public ontologies may want to hide some axioms to capitalize on their formalization efforts. Several approaches have been proposed in order to tackle the confidentiality requirements that arise form these scenarios. The most popular security criterion is that the published view of the knowledge base should not entail a secret sentence. However, there exist attacks that cannot be prevented this way. The user may exploit various sources of background knowledge and metaknowledge to reconstruct the hidden part of the knowledge base. This paper contributes to the area of knowledge base confidentiality in several ways:

(i) It highlights some vulnerabilities of the approaches that can be found in the literature, (Sec. 3).

(ii) It introduces a stronger confidentiality model that takes both object-level and meta-level background knowledge into account (Sec. 4), and it defines a method for computing secure knowledge views (Sec. 5) that generalizes some previous approaches.

(iii) It proposes a safe approximation of background metaknowledge (Sec. 6 and 7).

(iv) It investigates the computational complexity of constructing secure knowledge base views with our methodology (Sec. 7).

(v) It describes a prototypical implementation of the new framework (Sec. 9)

The paper is closed by a discussion of related work (Sec. 10), and conclusions. Proofs are omitted due to space limitations.

2 Preliminaries on Description Logics

We assume the reader to be familiar with description logics, and refer to [1] for all definitions and results. We assume a fixed, denumerable signature Σ specifying the names of *concepts*, *roles*, and *individuals*. Our framework is compatible with any description logic DL that enjoys compactness (needed by Theorem 6) and has decidable reasoning problems (e.g., $\mathcal{ALC}$, $\mathcal{EL}$, $\mathcal{SHIQ}$, etc.). We simply assume that our reference logical language $\mathcal{L}$ is generated from Σ by the grammar of the selected logic DL. By *axioms*, we mean members of $\mathcal{L}$, unless stated otherwise. A *knowledge base* is any subset of $\mathcal{L}$.[1]

Recall that axioms are expressions of the form $C \sqsubseteq D$, $R \sqsubseteq S$, $C(a)$, and $R(a,b)$ where C, D are concept expressions, R, S are role expressions, and a, b are individual constants. In some DL, an individual constant a may occur also in a *nominal*, that is, a concept expression $\{a\}$ denoting the singleton containing a. The axioms involving $\sqsubseteq$ are called *inclusions* (or *subsumptions*), while $C(a)$ and $R(a,b)$ are called *assertions*. In the simplest case, C and R are first order predicates and assertions are actually standard first-order atomic formulae. Inclusions are syntactic variants of logical implications.

The notion of *logical consequence* is the classical one; for all $K \subseteq \mathcal{L}$, the logical consequences of K will be denoted by $Cn(K)$ ($K \subseteq Cn(K) \subseteq \mathcal{L}$).

3 A simple confidentiality model

The most natural way of preserving confidentiality in a knowledge base *KB* is checking that its answers to user queries do not entail any secret. Conceptually, the queries of a user u are answered using u's view KB_u of the knowledge base, where KB_u is a maximal subset of *KB* that entails no secret. In order to illustrate some possible attacks to this mechanism, let us formalize the above *simple confidentiality model* (SCM).[2] It consists of: the knowledge base *KB* ($KB \subseteq \mathcal{L}$); a set of users U; a *view* $KB_u \subseteq KB$ for each $u \in U$; a set of *secrecies* $S_u \subseteq \mathcal{L}$ for each $u \in U$. Secrecies are axioms that may or may not be entailed by *KB*; if they do, then they are called *secrets* and must not be disclosed to u. Revealing that a secrecy is *not* entailed by *KB* is harmless, cf. [4].

A view KB_u is *secure* iff $Cn(KB_u) \cap S_u = \emptyset$. A view KB_u is *maximal secure* if it is secure and there exists no K such that $KB_u \subset K \subseteq KB$ and $Cn(K) \cap S_u = \emptyset$.

Attacks using object-level background knowledge. Frequently, part of the domain knowledge is not axiomatized in *KB*, therefore checking that $Cn(KB_u) \cap S_u = \emptyset$ does not suffice in practice to protect confidentiality. For example, suppose that there is one secret $S_u = \{\mathit{OncologyPatient}(\mathit{John})\}$ and $KB_u = \{\mathit{SSN}(\mathit{John}, \mathit{12345}), \mathit{SSN}(\mathit{user123}, \mathit{12345}), \mathit{OncologyPatient}(\mathit{user123})\}$. KB_u does not entail *OncologyPatient*(*John*), so according to the SCM model KB_u is secure. However, it is common knowledge that a SSN uniquely identifies a person, then the user can infer that $\mathit{John} = \mathit{user123}$, and hence the secret.

In other examples, the additional knowledge used to infer secrets may be stored in a public ontology or RDF repository, and confidentiality violations may be automated.

Attacks to complete knowledge. Suppose the attacker knows that *KB* has complete knowledge about a certain set of axioms. Then the attacker may be able to reconstruct some secrets from the "I don't know" answers of a maximal secure view KB_u.

Example 1. Consider a company's knowledge base that defines a concept *Employee* and a role *works_for* that describes which employees belong to which of the n departments

[1] Real knowledge bases are finite, but this restriction is not technically needed until Sec. 7.

[2] This usage of term "*model*" is common in Security & Privacy.

of the company, $d_1, \ldots, d_n$. The *KB* consists of assertions like:

$$\mathit{Employee}(e) \qquad (1) \qquad\qquad \mathit{works\_for}(e, d_i) \qquad (2)$$

where we assume that each employee e belongs to exactly one department d_i. A user u is authorized to see all assertions but the instances of (2) with $i = n$, because d_n is a special department, devoted to controlling the other ones. So S_u (the set of secrecies for u) is the set of all assertions $\mathit{works\_for}(e, d_n)$.

Note that there is one maximal secure view KB_u. It consists of all instances of (1), plus all instances of (2) such that $i \neq n$. Clearly, KB_u is secure according to SCM (because $Cn(KB_u) \cap S_u = \emptyset$). However, observe that $\mathit{works\_for}(e, d_n) \in Cn(KB)$ iff $\mathit{Employee}(e) \in Cn(KB_u)$ and for all $i = 1, \ldots, n$, $\mathit{works\_for}(e, d_i) \notin Cn(KB_u)$ (that is, the members of d_n are all the employees that apparently work for no department). Using this property (based on the knowledge that for each employee e, *KB* contains exactly one assertion $\mathit{works\_for}(e, d_i)$) and the knowledge of the protection mechanism (i.e. maximal secure views), that we assume to be known by attackers by *Kerchoff's principle*, a smart user can easily identify all the members of d_n. □

In practice, it is not hard to identify complete knowledge. A hospital's *KB* is expected to have complete knowledge about which patients are in which ward; a company's *KB* is likely to encode complete information about its employees, etc.

Some approaches filter query answers rather than publishing a subset of *KB* [8, 13, 15]. We call our abstraction of this method *simple answer confidentiality model* (SACM). It is obtained from the SCM by replacing the views $KB_u \subseteq KB$ with *answer views* $KB_u^a \subseteq Cn(KB)$. The difference is that KB_u^a is not required to be a subset of *KB* and—conceptually—KB_u^a may be infinite. KB_u^a is *secure* iff $Cn(KB_u^a) \cap S_u = \emptyset$.

The reader may easily verify that the SACM is vulnerable to the two kinds of attacks illustrated for the SCM. It is also vulnerable to a third kind of attacks, illustrated below.

Attacks to the signature. Suppose the user knows the signature of *KB* well enough to identify a symbol σ that does not occur in *KB*. First assume that σ is a concept name. It can be proved that:

Proposition 1. *If KB_u^a is a maximal secure answer view and σ is a concept name not occurring in KB, then for all secrecies $C \sqsubseteq D \in S_u$, $KB_u^a \models C \sqcap \sigma \sqsubseteq D$ iff $KB \models C \sqsubseteq D$.*

The problem is that although $C \sqcap \sigma \sqsubseteq D$ does not entail the secret inclusion $C \sqsubseteq D$, still a smart user knows that the former inclusion cannot be proved unless *KB* entails also the latter (then maximal secure answer views generally fail to protect secrets). This attack can be easily adapted to the case where σ is a role name. In practice, it is not necessary to be sure that σ does not occur in *KB*. The attacker may make a sequence of educated guesses (say, by trying meaningless long strings, or any word that is clearly unrelated to the domain of the *KB*); after a sufficient number of trials, the majority of answers should agree with the "real" answer with high probability. Rejecting queries whose signature is not contained in *KB*'s signature mitigates this kind of attacks but it leaks *KB*'s signature and it does not provide a complete solution. The attacker may still guess a σ which is logically unrelated to C and D and carry out a similar attack.

4 A meta-safe confidentiality model

In this section we introduce a confidentiality model that makes the vulnerabilities illustrated above visible, by taking into account object- and meta-level background knowledge. A *bk-model* $\mathcal{M} = \langle KB, U, f, \langle S_u, PKB_u, BK_u \rangle_{u \in U} \rangle$ consists of a knowledge base $KB \subseteq \mathcal{L}$, a set of users U, plus:

- a *filtering function* $f : \wp(\mathcal{L}) \times U \to \wp(\mathcal{L})$, mapping each knowledge base K and each user u on a view $f(K, u) \subseteq Cn(K)$;
- for all $u \in U$:
  - a finite set of secrecies $S_u \subseteq \mathcal{L}$;
  - a set of axioms $BK_u \subseteq \mathcal{L}$, encoding the users' object-level knowledge;
  - a set of *possible knowledge bases* $PKB_u \subseteq \wp(\mathcal{L})$ (users' metaknowledge).[3]

The view of KB released to a user u is $f(KB, u)$. We adopt PKB because at this stage we do not want to tie our framework to any specific metalanguage. PKB represents the knowledge bases that are compatible with the user's metaknowledge.

Definition 1. *A filtering function f is* secure *(w.r.t. $\mathcal{M}$) iff for all $u \in U$ and all $s \in S_u$, there exists $K \in PKB_u$ such that:*
1. *$f(K, u) = f(KB, u)$;*
2. *$s \notin Cn(K \cup BK_u)$.*

Intuitively, if f is safe according to Def. 1, then no user u can conclude that any secret s is entailed by the KB she is interacting with—enhanced with the object-level background knowledge BK_u—for the following reasons: By point 1, KB and K have the same observable behavior, and K is a possible knowledge base for u since $K \in PKB_u$; therefore, as far as u knows, the knowledge base might be K. Moreover, by point 2, K and the object-level background knowledge BK_u do not suffice to entail the secret s.

In the rest of the paper we tacitly assume that no secret is violated a priori, that is, for all secrets $s \in S_u$ there exists $K \in PKB_u$ such that $s \notin Cn(K \cup BK_u)$.[4] *Moreover, in order to improve readability, we shall omit the user u from subscripts and argument lists* whenever u is irrelevant to the context.

The attacks discussed in Section 3 can be easily formalized in this setting; so, in general, the maximal secure views of SCM are not secure according to Def. 1.

Example 2. Example 1 can be formalized in our model as follows: The set of secrecies S is the set of all assertions $works\_for(e, d_n)$; $BK = \emptyset$ and PKB is the set of all the knowledge bases K that consist of assertions like (1) and (2), and such that for each axiom $Employee(e)$, K contains exactly one corresponding axiom $works\_for(e, d_i)$ and viceversa. The filtering function f maps each $K \in PKB$ on the maximal subset of K that entails none of S's members, that is, $f(K) = K \setminus S$ (by definition of PKB).

Note that f is injective over PKB, so condition 1 of Def. 1 is satisfied only if $K = KB$. So, if KB contains at least one secret, then the conditions of Def. 1 cannot be satisfied, that is, maximal secure SCM views are not secure in our model. Indeed, KB can be

[3] In practice, bk-models are finite, and filterings computable, but no such assumption will be technically needed until Sec. 7.

[4] Conversely, no filtering function can conceal a secret that is already known by the user.

reconstructed from the secure view by observing that $KB = f(KB) \cup \{works\_for(e, d_n) \mid Employee(e) \in f(KB) \wedge \forall i = 1, \dots, n, works\_for(e, d_i) \notin f(KB)\}$. □

Similarly, the formalizations of the other attacks yield injective filtering functions (the details are left to the reader).

5 A meta-secure query answering mechanism

In this section we introduce a *secure filtering function*. It is formulated as an iterative process based on a *censor*, that is a boolean function that decides for each axiom whether it should be obfuscated to protect confidentiality. The iterative construction manipulates pairs $\langle X^+, X^- \rangle \in \wp(\mathcal{L}) \times \wp(\mathcal{L})$ that represent a meta constraint on possible knowledge bases: we say that a knowledge base K *satisfies* $\langle X^+, X^- \rangle$ iff K entails all the sentences in X^+ and none of those in X^- (formally, $Cn(K) \supseteq X^+$ and $Cn(K) \cap X^- = \emptyset$).

Let *PAX* (the set of *possible axioms*) be the set of axioms that may occur in the knowledge base according to the user's knowledge, i.e. $PAX = \bigcup_{K' \in PKB} K'$. Let $\nu = |PAX| + 1$ if *PAX* is finite and $\nu = \omega$ otherwise; let $\alpha_1, \alpha_2, \dots, \alpha_i, \dots$ be any enumeration of *PAX* ($i < \nu$).[5] The secure view construction for a knowledge base K in a bk-model $\mathcal{M}$ consists of the following, inductively defined sequence of pairs $\langle K_i^+, K_i^- \rangle_{i \geq 0}$:

- $\langle K_0^+, K_0^- \rangle = \langle \emptyset, \emptyset \rangle$, and for all $1 \leq i < \nu$, $\langle K_{i+1}^+, K_{i+1}^- \rangle$ is defined as follows:
  - if $censor_{\mathcal{M}}(K_i^+, K_i^-, \alpha_{i+1}) = true$ then let $\langle K_{i+1}^+, K_{i+1}^- \rangle = \langle K_i^+, K_i^- \rangle$;
  - if $censor_{\mathcal{M}}(K_i^+, K_i^-, \alpha_{i+1}) = false$ and $K \models \alpha_{i+1}$ then $\langle K_{i+1}^+, K_{i+1}^- \rangle = \langle K_i^+ \cup \{\alpha_{i+1}\}, K_i^- \rangle$;
  - otherwise let $\langle K_{i+1}^+, K_{i+1}^- \rangle = \langle K_i^+, K_i^- \cup \{\alpha_{i+1}\} \rangle$.

Finally, let $K^+ = \bigcup_{i<\nu} K_i^+$, $K^- = \bigcup_{i<\nu} K_i^-$, and $f_{\mathcal{M}}(K, u) = K^+$.

Note that the inductive construction aims at finding maximal sets K^+ and K^- that (i) partly describe what does / does not follow from K (as K satisfies $\langle K^+, K^- \rangle$ by construction), and (ii) do not trigger the censor (the sentences α_{i+1} that trigger the censor are included neither in K^+ nor in K^-, cf. the induction step).

In order to define the censor we need an auxiliary definition that captures all the sentences that can be entailed with the background knowledge *BK* and the meta-knowledge *PKB* enriched by a given constraint $\langle X^+, X^- \rangle$ analogous to those adopted in the iterative construction: Let $Cn_{\mathcal{M}}(X^+, X^-)$ be the set of all axioms $\alpha \in \mathcal{L}$ such that

$$\text{for all } K' \in PKB \text{ such that } K' \text{ satisfies } \langle X^+, X^- \rangle, \alpha \in Cn(K' \cup BK) . \tag{3}$$

Now the censor is defined as follows: For all $X^+, X^- \subseteq \mathcal{L}$ and $\alpha \in \mathcal{L}$,

$$censor_{\mathcal{M}}(X^+, X^-, \alpha) = \begin{cases} true & \text{if there exists } s \in S \text{ s.t. either } s \in Cn_{\mathcal{M}}(X^+ \cup \{\alpha\}, X^-) \\ & \text{or } s \in Cn_{\mathcal{M}}(X^+, X^- \cup \{\alpha\}); \\ false & \text{otherwise.} \end{cases} \tag{4}$$

In other words, the censor checks whether telling either that α is derivable or that α is not derivable to a user aware that the knowledge base satisfies $\langle X^+, X^- \rangle$, restricts the

[5] We will show later how to restrict the construction to finite sequences, by approximating *PAX*.

set of possible knowledge bases enough to conclude that a secret s is entailed by the knowledge base and the background knowledge BK.

Note that the censor obfuscates α_{i+1} if *any* of its possible answers entail a secret, independently of the actual contents of K (the two possible answers "yes" and "no" correspond to conditions $s \in Cn_{\mathcal{M}}(X^+ \cup \{\alpha\}, X^-)$ and $s \in Cn_{\mathcal{M}}(X^+, X^- \cup \{\alpha\})$, respectively). In this way, roughly speaking, the knowledge bases that entail s are given the same observable behavior as those that don't. Under a suitable continuity assumption on $Cn_{\mathcal{M}}$, this enforces confidentiality:

Theorem 1. *If $Cn_{\mathcal{M}}(KB^+, KB^-) \subseteq \bigcup_{i<\nu} Cn_{\mathcal{M}}(KB_i^+, KB_i^-)$, then $f_{\mathcal{M}}$ is secure w.r.t. $\mathcal{M}$.*

Examples of the behavior of $f_{\mathcal{M}}$ are deferred until Sec.7.

6 Approximating background knowledge

Of course, the actual confidentiality of a filtering $f(KB, u)$ depends on a careful definition of the user's background knowledge, that is, PKB_u and BK_u. If background knowledge is not exactly known, as it typically happens, then it can be safely approximated by *overestimating* it. More background knowledge means larger BK_u and smaller PKB_u, which leads to the following comparison relation $\leq_k$ over bk-models:

Definition 2. *Given two bk-models $\mathcal{M} = \langle KB, U, f, \langle S_u, PKB_u, BK_u \rangle_{u \in U} \rangle$ and $\mathcal{M}' = \langle KB', U', f', \langle S'_u, PKB'_u, BK'_u \rangle_{u \in U'} \rangle$, we write $\mathcal{M} \leq_k \mathcal{M}'$ iff*

1. *$KB = KB'$, $U = U'$, $f = f'$, and $S_u = S'_u$ (for all $u \in U$);*
2. *for all $u \in U$, $PKB_u \supseteq PKB'_u$ and $BK_u \subseteq BK'_u$.*

The next proposition proves that a bk-model $\mathcal{M}$ can be safely approximated by any $\mathcal{M}'$ such that $\mathcal{M} \leq_k \mathcal{M}'$:

Proposition 2. *If f is secure w.r.t. $\mathcal{M}'$ and $\mathcal{M} \leq_k \mathcal{M}'$, then f is secure w.r.t. $\mathcal{M}$.*

Consequently, a generic advice for estimating BK consists in including as many pieces of relevant knowledge as possible, for example:

(i) modelling as completely as possible the integrity constraints satisfied by the data, as well as role domain and range restrictions and disjointness constraints;

(ii) including in BK all the relevant public sources of formalized relevant knowledge (such as ontologies and triple stores).

While object-level background knowledge is dealt with in the literature, the general metaknowledge encoded by PKB is novel. Therefore, the next section is focussed on some concrete approximations of PKB and their properties.

7 Approximating and reasoning about possible knowledge bases

In this section, we investigate the real world situations where *the knowledge base KB is finite* and *so are all the components of bk-models* (U, S_u, BK_u, PKB_u); then we focus on PKB_u that contain only finite knowledge bases. Consequently, $f_{\mathcal{M}}$ will turn out to be decidable and we will study its complexity under different assumptions.

A language for defining *PKB* is a necessary prerequisite for the practical implementation of our framework and a detailed complexity analysis of $f_{\mathcal{M}}$. Here we express *PKB* as the set of all theories that are contained in a given set of *possible axioms PAX* [6] and satisfy a given, finite set *MR* of *metarules* like:

$$\alpha_1, \ldots, \alpha_n \Rightarrow \beta_1 \mid \ldots \mid \beta_m \quad (n \geq 0, m \geq 0)\,, \tag{5}$$

where all α_i and β_j are in $\mathcal{L}$ ($1 \leq i \leq n$, $1 \leq j \leq m$). Informally, (5) means that if *KB* entails $\alpha_1, \ldots, \alpha_n$ then *KB* entails also some of $\beta_1, \ldots, \beta_m$. Sets of similar metarules can be succintly specified using *metavariables*; they can be placed wherever individual constants may occur, that is, as arguments of assertions, and in nominals. A metarule with such variables abbreviates the set of its *ground instantiations*: Given a $K \subseteq \mathcal{L}$, let $\mathit{ground}_K(\mathit{MR})$ be the ground (variable-free) instantiation of *MR* where metavariables are uniformly replaced by the individual constants occurring in *K* in all possible ways.

Example 3. Let $\mathit{MR} = \Big\{ \exists R.\{X\} \Rightarrow A(X) \Big\}$, where X is a metavariable, and let $K = \Big\{ R(a,b) \Big\}$. Then $\mathit{ground}_K(\mathit{MR}) = \Big\{ (\exists R.\{a\} \Rightarrow A(a)),\ (\exists R.\{b\} \Rightarrow A(b)) \Big\}$. □

If r denotes rule (5), then let $\mathit{body}(r) = \{\alpha_1, \ldots, \alpha_n\}$ and $\mathit{head}(r) = \{\beta_1, \ldots, \beta_m\}$. We say r is *Horn* if $|\mathit{head}(r)| \leq 1$. A set of axioms $K \subseteq \mathcal{L}$ *satisfies* a ground metarule r if either $\mathit{body}(r) \nsubseteq \mathit{Cn}(K)$ or $\mathit{head}(r) \cap \mathit{Cn}(K) \neq \emptyset$. In this case we write $K \models_m r$.

Example 4. Let A, B, C be concept names and R be a role name. The axiom set $K = \{A \sqsubseteq \exists R.B, A \sqsubseteq C\}$ satisfies $A \sqsubseteq \exists R \Rightarrow A \sqsubseteq B \mid A \sqsubseteq C$ but not $A \sqsubseteq \exists R \Rightarrow A \sqsubseteq B$. □

Moreover, if K satisfies all the metarules in $\mathit{ground}_K(\mathit{MR})$ then we write $K \models_m \mathit{MR}$. Therefore the formal definition of *PKB* now becomes:

$$\mathit{PKB} = \{K \mid K \subseteq \mathit{PAX} \wedge K \models_m \mathit{MR}\}\,. \tag{6}$$

In accordance with Prop. 2, we approximate *PAX* in a conservative way. We will analyze two possible definitions:

1. $\mathit{PAX}_0 = \mathit{KB}$ (i.e., as a minimalistic choice we only assume that the axioms of *KB* are possible axioms; of course, by Prop. 2, this choice is safe also w.r.t. any larger *PAX* where *at least* the axioms of *KB* are regarded as possible axioms);
2. $\mathit{PAX}_1 = \mathit{KB} \cup \bigcup_{r \in \mathit{ground}_{\mathit{KB}}(\mathit{MR})} \mathit{head}(r)$.

Remark 1. The latter definition is most natural when metarules are automatically extracted from *KB* with rule mining techniques, that typically construct rules using material from the given *KB* (then rule heads occur in *KB*).

Example 5. Consider again Example 1. The user's metaknowledge about *KB*'s completeness can be encoded with:

$$\mathit{Employee}(X) \Rightarrow \mathit{works\_for}(X, d_1) \mid \ldots \mid \mathit{works\_for}(X, d_n)\,, \tag{7}$$

[6] Differently from Sec. 5, here *PKB* is defined in terms of *PAX*.

where X is a metavariable. First let $PAX = PAX_1$. The secure view $f_{\mathcal{M}}(KB)$ depends on the enumeration order of PAX. If the role assertions $works\_for(e, d_i)$ precede the concept assertions $Employee(e)$, then, in a first stage, the sets KB^+_j are progressively filled with the role assertions with $d_i \neq d_n$ that belong to KB, while the sets KB^-_j accumulate all the role assertions that do not belong to KB. In a second stage, the sets KB^+_j are further extended with the concept assertions $Employee(e)$ such that e does not work for d_n. The role assertions $works\_for(e, d_n)$ of KB and the corresponding concept assertions $Employee(e)$ are neither in KB^+ nor in KB^-. Note that the final effect is equivalent to removing from KB all the axioms referring to the individuals that work for d_n. Analogously, in [8] the individuals belonging to a specified set are removed from all answers.

Next suppose that the role assertions $works\_for(e, d_i)$ follow the concept assertions $Employee(e)$, and that each $works\_for(e, d_i)$ follows all $works\_for(e, d_k)$ such that $k < i$. Now all the assertions $Employee(e)$ of KB enter KB^+, and all axioms $works\_for(e, d_i)$ with $i < n - 1$ enter either KB^+ or KB^-, depending on whether they are members of KB or not. Finally, the assertions $works\_for(e, d_i) \in Cn(KB)$ with $i \in \{n - 1, n\}$ are inserted neither in KB^+ nor in KB^-, because the corresponding instance of (7) with $X = e$ has the body in KB^+ and the first $n - 2$ alternatives in the head in KB^-, therefore a negative answer to $works\_for(e, d_{n-1})$ would entail the secret $works\_for(e, d_n)$ by (7). This triggers the censor for all assertions $works\_for(e, d_{n-1})$. Summarizing, with this enumeration ordering it is possible to return the complete list of employees; the members of d_n are protected by hiding also which employees belong to d_{n-1}.

Finally, let $PAX = PAX_0$. Note that in this case all possible knowledge bases are subsets of KB, that contains exactly one assertion $works\_for(e, d_{i(e)})$ for each employee e. To satisfy (7), every $K \in PKB$ containing $Employee(e)$ must contain also $works\_for(e, d_{i(e)})$. It follows that $f_{\mathcal{M}}$ must remove all references to the individuals that work for d_n, as it happens with the first enumeration of PAX_1. □

Definition 3. *A bk-model $\mathcal{M}$ is* canonical *if for all users $u \in U$, PAX_u is either PAX_0 or PAX_1 and PKB_u is defined by (6) for a given MR_u. Moreover, $\mathcal{M}$ is* in a description logic DL *if for all $u \in U$, all the axioms in KB, PKB_u, BK_u, and S_u belong to DL.*

The size of PAX_0 and PAX_1 is polynomial in the size of $KB \cup MR$, therefore PKB is finite and exponential in the size of $KB \cup MR$. Finiteness implies the continuity hypothesis on $Cn_{\mathcal{M}}$ of Theorem 1, and hence (using Theorem 1 and Prop. 2):

Theorem 2. *If $\mathcal{M}$ is canonical, then $f_{\mathcal{M}}$ is secure with respect to all $\mathcal{M}' \leq_k \mathcal{M}$.*

First we analyze the complexity of constructing the secure view $f_{\mathcal{M}}(KB)$ when the underlying description logic is tractable, like $\mathcal{EL}$ and DL-lite for example.

Lemma 1. *If the axioms occurring in MR and K are in a DL with tractable subsumption and instance checking, then checking $K \models_m MR$ is:*

1. *in P if either MR is ground or there exists a fixed bound on the number of distinct variables in MR;*
2. *coNP-complete otherwise.*

With Lemma 1, one can prove the following two lemmas.

Lemma 2. *Let $\mathcal{M}$ range over canonical bk-models. If $\mathcal{M}$, s, X^+, and X^- are in a DL with tractable subsumption/instance checking, and the number of distinct variables in MR is bounded by a constant, then checking whether $s \in Cn_{\mathcal{M}}(X^+, X^-)$ is:*

1. *in P if MR is Horn and $PAX = PAX_1$;*
2. *coNP-complete if either MR is not Horn or $PAX = PAX_0$.*

Lemma 3. *Let $\mathcal{M}$ be a canonical bk-model. If $\mathcal{M}$, s, X^+, and X^- are in a DL with tractable entailment problems, and there is no bound on the number of variables in the metarules of MR, then checking $s \in Cn_{\mathcal{M}}(X^+, X^-)$ is:*

1. *in P^{NP} if MR is Horn and $PAX = PAX_1$;*
2. *in Π_2^p if either MR is not Horn or $PAX = PAX_0$.*

The value of $censor(X^+, X^-, \alpha)$ can be computed straightforwardly by iterating the tests $s \in Cn_{\mathcal{M}}(X^+ \cup \{\alpha\}, X^-)$ and $s \in Cn_{\mathcal{M}}(X^+, X^- \cup \{\alpha\})$ for all secrets $s \in S$. Since the set of secrets is part of the parameter $\mathcal{M}$ of the filtering function, the number of iterations is polynomial in the input and the complexity of the censor is dominated by the complexity of $Cn_{\mathcal{M}}()$. The latter is determined by Lemma 2 and Lemma 3, so we immediately get:

Corollary 1. *Let $\mathcal{M}$ be a canonical bk-model and suppose that $\mathcal{M}$, X^+, X^-, and α are in a DL with tractable entailment problems. If the number of distinct variables in MR is bounded by a constant, then computing $censor(X^+, X^-, \alpha)$ is:*

- *in P if MR is Horn and $PAX = PAX_1$;*
- *coNP-complete if either MR is not Horn or $PAX = PAX_0$.*

If there is no bound on the number of variables in the metarules of MR, then computing $censor(X^+, X^-, \alpha)$ is:

- *in P^{NP} if MR is Horn and $PAX = PAX_1$;*
- *in Π_2^p if either MR is not Horn or $PAX = PAX_0$.*

We are now ready to analyze the complexity of filtering functions:

Theorem 3. *If $\mathcal{M}$ is a canonical bk-models in a DL with tractable entailment problems, then computing $f_{\mathcal{M}}(KB)$ is:*

1. *P-complete if the number of distinct variables in the rules of MR is bounded, MR is Horn, and $PAX = PAX_1$;*
2. *P^{NP}-complete if the number of distinct variables in MR is bounded, and either MR is not Horn or $PAX = PAX_0$;*
3. *in P^{NP} if the variables in MR are unbounded, MR is Horn, and $PAX = PAX_1$;*
4. *in Δ_3^p if MR is not restricted and $PAX \in \{PAX_0, PAX_1\}$.*

Theorem 4. *Computing $f_{\mathcal{M}}(KB)$ over canonical $\mathcal{M}$ in a DL with ExpTime entailment (e.g. $\mathcal{ALCQO}$, $\mathcal{ALCIO}$, $\mathcal{ALCQI}$, $\mathcal{SHOQ}$, $\mathcal{SHIO}$, $\mathcal{SHIQ}$), is still in ExpTime.*

Theorem 5. *Computing $f_{\mathcal{M}}(KB)$ over canonical $\mathcal{M}$ in $\mathcal{SROIQ}(\mathcal{D})$ is in $coNP^{N2ExpTime}$.*

8 Relationships with the SCM

Here we show that the meta-secure framework is a natural generalization of the SCM. The main result—roughly speaking—demonstrates that the SCM model can be essentially regarded as a special case of our framework where $PKB \supseteq \wp(KB)$ and $BK = \emptyset$. In this case $f_{\mathcal{M}}$ is secure even if $\mathcal{M}$ is not assumed to be canonical.

Theorem 6. *Let $\mathcal{M} = \langle KB, U, f_{\mathcal{M}}, \langle S_u, PKB_u, BK_u \rangle_{u \in U} \rangle$. If $PKB = \wp(KB)$, $BK = \emptyset$, and KB is finite, then*

1. $Cn_{\mathcal{M}}(KB^+, KB^-) = \bigcup_{i<\nu} Cn_{\mathcal{M}}(KB_i^+, KB_i^-)$.
2. *For all enumerations of PAX, the corresponding $f_{\mathcal{M}}(KB, u)$ is logically equivalent to a maximal secure view KB_u of KB according to the SCM; conversely, for all maximal secure view KB_u of KB (according to the SCM) there exists an enumeration of PAX such that the resulting $f_{\mathcal{M}}(KB, u)$ is logically equivalent to KB_u.*
3. *$f_{\mathcal{M}}$ is secure w.r.t. $\mathcal{M}$ and w.r.t. any $\mathcal{M}' = \langle KB, U, f_{\mathcal{M}}, \langle S_u, PKB'_u, BK'_u \rangle_{u \in U} \rangle$ such that $PKB' \supseteq \wp(KB)$ and $BK' = \emptyset$.*

Theorem 6 applies to every canonical $\mathcal{M}$ such that $MR = BK = \emptyset$, because $MR = \emptyset$ implies that $PAX_0 = PAX_1 = KB$ and hence $PKB = \wp(KB)$. This shows that the SCM can be regarded as a special case of our framework where the user has no background knowledge. Moreover, by this correspondence, one immediately obtains complexity bounds for the SCM from those for PAX_1 and Horn, bounded-variable *MR*.

9 Framework Implementation

In this section we introduce a prototypical implementation of the framework based on PAX_1 and Horn metarules. Nowadays ontologies are managed with the help of the OWL API[7] and DL reasoners which allow us to take full advantage of their rich underlying semantics. Unfortunately, the OWL reasoners publicly available do not offer native support for conjunctive query answering required to process users' metaknowledge. A partial exception of this rule is the Pellet reasoner discussed later. Straightforward evaluation of metarules in the presence of metavariables with an OWL reasoner would need to consider all possible ways of uniformly replacing metavariables by individual constants occurring in the ontology. On the other hand, the evaluation of a ground rule r with an OWL reasoner in the worst case would require checking that all the axioms $\alpha_1, \ldots, \alpha_n \in body(r)$ and $\beta_1, \ldots, \beta_m \in head(r)$ are entailed by *KB*. Summing up, with this method, as the ontology ABox grow, metarule evaluation can easily become unmanageable in terms of execution time. Consequently, the presence of technologies that permit native conjunctive query evaluation reveals fundamental to achieve efficient implementation of the framework. SPARQL[8], the W3C standard that provide languages and protocols to query and manipulate RDF content (and so ontologies encoded in the XML/RDF Syntax), constitute a de facto standard when it comes to conjunctive query answering. It has recently been extended with the so-called entailment regimes, which

[7] http://owlapi.sourceforge.net/

[8] http://www.w3.org/TR/sparql11-overview/

define how queries are evaluated under more expressive semantics, such as OWL semantics, than the SPARQL standard simple entailment, based essentially on pattern matching on graphs. Unfortunately, most of the available engines do not provide support for OWL reasoning. To the best of our knowledge only Apache Jena Semantic Web Toolkit and Pellet support OWL inference over the queried ontologies. Moreover, Pellet query engine seems not to have been reengineered for the last few years. It was therefore an obvious option to prefer the Jena query engine for our system. The Jena inference subsystem is designed to allow usage of a number of predefined Jena OWL reasoners, as well as external reasoners[9]. However, the usage of the internal reasoners is recommended for efficiency reasons. Note, that OWL, OWL Mini, OWL Micro Jena Reasoners are a set of useful but not a full-fledged rule-based implementations of the OWL/Lite subset of the OWL/Full language. Critical not supported constructs which go beyond OWL/Lite are *complementOf* and *oneOf*, while the support for *unionOf* is partial. We consider our choice to use a Jena OWL reasoner a good compromise between expressivity and efficiency. According to the theoretical framework the system consists of two modules. The first one actuates the parsing of the user's metaknowledge represented by means of a set of metarules. The second module is in charge of the secure ontology view construction.

Algorithm 1 provides an abstract view on the implementation of our framework. It takes as input an ontology KB, a set of secrets S, a set of metarules MR and the user's background knowledge BK. The output is the set of axioms that constitute a secure ontology view for the user. The set M_M and M_G form a partition of MR according to rule types (ground or containing metavariables).

Remark 2. By standard logic programming techniques, a minimal $PKB \subseteq PAX_1$ satisfying the set of metarules and the constraints K^+ can be obtained with the following PTIME construction:

$$PKB_0 = K^+ , \quad PKB_{i+1} = PKB_i \cup \bigcup \{ head(r) \mid r \in ground_{PKB_i}(MR) \wedge body(r) \subseteq Cn(PKB_i) \}$$

The sequence's limit $PKB_{|PAX_1|}$ satisfy $\langle K^+, K^- \rangle$ as well if $Cn(PKB_{|PAX_1|}) \cap K^- \neq \emptyset$. Then, for all $s \in S$, $s \in Cn_{\mathcal{M}}(K^+, K^-)$ holds iff $s \in Cn(PKB_{|PAX_1|} \cup BK)$. For more details refer to [7].

By iterating over the axioms $\alpha \in PAX_1$ (*line 6-25*), PKB collects at each step all parts of PAX_1 that can be revealed to the user. The repeat-until loop (*lines 9-17*) computes the deductive closure $PKB^{'}$ of PKB under MR[10]. In particular, for every ground metarule (*lines 10-13*) we execute a SPARQL ASK query (hidden in *line 11*) to verify if its body is entailed by the current $PKB^{'}$. For every metarule containing metavariables (*lines 14-16*) we execute a SPARQL SELECT query (encoded in *line 15*) in order to obtain all possible bindings of the metavariables that satisfy the metarule's body. The pair of steps described above is iterated until a fixpoint is reached (no elements are added to $PKB^{'}$ (*line 17*)). At this point the condition $Cn(PKB^{'}) \cap K_i^- \neq \emptyset$ is checked (*line 18*). We are now ready to determine the value of the censor function for α. We verify that no secret is entailed from the minimal PKB (*line 19*) taking in consideration

[9] To be plugged into Jena a reasoner must expose Jena API.

[10] The result of Proposition 2 guarantees that considering only the minimal PKB is sound.

Algorithm 1:

```
   Data: $KB, S, MR, BK$.
1  $K_i^+, K_i^- \leftarrow \emptyset$;
2  $M_M \leftarrow \{r_i | r_i \in MR$ and $r_i$ metarule containing metavariables$\}$;
3  $M_G \leftarrow \{r_i | r_i \in MR$ and $r_i$ ground metarule$\}$;
4  $PAX_1 \leftarrow \{\alpha \in KB \cup \bigcup_{r \in ground_{KB}(MR)} head(r)\}$;
5  $PKB \leftarrow \emptyset$;
6  forall $\alpha \in PAX_1$ do
7      $PKB' \leftarrow PKB \cup \{\alpha\}$;
8      $M'_G \leftarrow M_G$;
9      repeat
10         forall $m \in M'_G$ do
11             if $PKB' \models body(m)$ then
12                 $PKB' \leftarrow PKB' \cup \{head(m)\}$;
13                 $M'_G \leftarrow M'_G \setminus \{m\}$;
14         forall $m \in M_M$ do
15             forall $(a_0, \ldots, a_n) \mid PKB' \models body(m, [X_0/a_0, \ldots, X_n/a_n])$ do
16                 $PKB' \leftarrow PKB' \cup \{head(m, [X_0/a_0, \ldots, X_n/a_n])\}$;
17     until No element is added to $PKB'$;
18     if $\{\beta \in K_i^- \mid PKB' \models \beta\} = \emptyset$ then
19         if $\{s \in S \mid PKB' \cup BK \models s\} = \emptyset$ then
20             if $KB \models \alpha$ then
21                 $K_i^+ \leftarrow K_i^+ \cup \{\alpha\}$;
22                 $PKB \leftarrow PKB'$;
23                 $M_G \leftarrow M'_G$;
24             else
25                 $K_i^- \leftarrow K_i^- \cup \{\alpha\}$;
26 return $K_i^+$
```

the background knowledge[11]. In case α is entailed by KB, it is safe to include it in the view (*line 21*). Otherwise, the set K_i^- is updated (*line 25*). Note that we need an OWL reasoner in order to perform the entailment checks in *lines 18-20*. We make use of the incremental reasoner Pellet, that for each α_i in the enumeration of PAX_1, is expected to restrict reasoning to the new inferences triggered by α without repeating the inferences that involve only K_{i-1}^+.

A first optimization regards the evaluation of the set of ground rules M_G. During the construction of *PKB*, due to the monotonicity of reasoning, at each iteration we can safely remove from M_G all the ground rules already satisfied at the previous iterations (*line 13,23*). Another optimization concerns the evaluation order of PAX_1. Checking $Cn(PKB_{|PAX_1|}) \cap K^- \neq \emptyset$ (*line 18*) is time consuming, so we adopt an approach that main-

[11] This corresponds to check whether $s \in Cn_{\mathcal{M}}(K^+ \cup \{\alpha\}, K^-)$ only. The condition $s \in Cn_{\mathcal{M}}(K^+, K^- \cup \{\alpha\})$ is in fact embedded in *line 18*.

tain the set K_i^- as small as possible. This is achieved processing all $\{\alpha \in PAX_1 \mid \alpha \in KB\}$ in *line 6* first. Provided that the condition in *line 19* is vacuously satisfied we are sure that the K_i^- remains empty.

Experimental analysis show that the generation of secure views for medium sized ontologies may take several minutes. We plan to investigate module extraction techniques that are expected to improve drastically the execution time by restricting the part of the knowledge base on which metarules apply.

10 Related work

Baader et al. [2], Eldora et al. [11], and Knechtel and Stuckenschmidt [13] attach security labels to axioms and users to determine which subset of the KB can be used by each subject. These works are instances of the SCM so they are potentially vulnerable to the attacks based on background knowledge; this holds in particular for [13] that pursues the construction of maximal secure views. Moreover, in [2, 11] axiom labels are not derived from the set of secrets; knowledge engineers are responsible for checking ex post that no confidential knowledge is entailed; in case of leakage, the view can be modified with a revision tool based on pinpointing. On the contrary, our mechanism automatically selects which axioms shall be hidden in order to produce a secure view.

Chen and Stuckenschmidt [8] adopt an instance of the SACM based on removing some individuals entirely. In general, this may be secure against metaknowledge attacks (cf. Ex. 5). However, no methodology is provided for selecting the individuals to be removed given a target set of secrets. In [3], *KB* is partitioned into a visible part KB_v and a hidden part KB_h. Conceptually, this is analogous to axiom labelling, cf. the above approaches. Their confidentiality methodology seems to work only under the assumption that the signatures of KB_v and KB_h are disjoint, because in strong safety they do not consider the formulae that are implied by a combination of KB_v and KB_h. Surely the axioms of KB_h whose signature is included in the signature of KB_v cannot be protected, in general. A partition-based approach is taken in [10], too. It is not discussed how to select the hidden part KB_h given a set of target secrets (which includes the issue of deciding secondary protection).

Similarly, in [14] only ex-post confidentiality verification methods are provided. In their model the equivalent of *PKB* is the set of all knowledge bases that include a given set of publicly known axioms $\mathcal{S} \subseteq KB$; consequently, in some cases their verification method is vulnerable to the attacks to complete knowledge, that are based on more complex (conditional) metaknowledge (cf. Example 2 and Example 5) that cannot be encoded in their framework.

Cuenca Grau and Horrocks [9] investigate knowledge confidentiality from a probabilistic perspective: enlarging the public view should not change the probability distribution over the possible answers to a “sensitive query” Q that represents the set of secrets. In [9] users can query the knowledge base only through a pre-defined set of views (we place no such restriction, instead). A probability distribution P over the set of knowledge bases plays a role similar to metaknowledge. However, their confidentiality condition allows P to be replaced with a different P' after enlarging the public view, so at a closer look P does not really model the user's a priori knowledge about the knowledge base (that should remain constant), differently from our *PKB*.

Our method is inspired by the literature on *controlled query evaluation* (CQE) based on lies and/or refusals ([4, 5, 6] etc). Technically we use *lies*, because rejected queries are not explicitly marked (the cited papers use the special answer "mum"). However, our censor resembles the classical refusal censor, so the properties of $f_{\mathcal{M}}$ are not subsumed by any of the classical CQE methods. For example (unlike the CQE approaches that use lies), $f_{\mathcal{M}}(KB, u)$ encodes only correct knowledge (i.e. entailed by *KB*), and it is secure whenever users do not initially know any secret (while lies-based CQE further require that no *disjunction* of secrets should be known a priori). Unlike the refusal method, $f_{\mathcal{M}}$ can handle *cover stories* because users are not told that some queries are obfuscated; as an additional advantage, our method needs not to adapt existing engines to handle nonstandard answers like *mum*. Finally, the CQE approaches do not deal specifically with DL knowledge bases, metaknowledge, and related complexity analysis.

11 Conclusions

The confidentiality preservation methods that do not consider background knowledge are vulnerable to several attacks. We identified two vulnerabilities (attacks to complete knowledge and to the signature) and introduced a knowledge base confidentiality model that can detect these vulnerabilities, based on a fully generic formalization of object- and meta-level background knowledge. Confidentiality is enforced through a generic mechanism for constructing secure views (the filtering $f_{\mathcal{M}}$) that is provably secure w.r.t. the meta-confidentiality model under a continuity assumption, and generalizes a few previous approaches (cf. Thm. 6 and Ex. 5). In order to compute secure views in practice we introduced a safe, generic method for approximating background knowledge, together with a specific rule-based language for expressing metaknowledge. Based on this instantiation of the general framework, where $f_{\mathcal{M}}$ is always secure, we analyzed the computational complexity of computing secure views. If the underlying DL is tractable, then in the simplest case $f_{\mathcal{M}}$ can be computed in polynomial time. The number of variables in metarules and the adoption of a more secure approximation (PAX_0) may increase complexity up to $\mathrm{P}^{NP} = \Delta_2^p$ and perhaps Δ_3^p. The complexity of non-Horn metarules, however, can be avoided by replacing each non-Horn r with one of its Horn strengthenings: $body(r) \Rightarrow \alpha$ such that $\alpha \in head(r)$. This approximation is safe (because it restricts *PKB*), and opens the way to a systematic use of the low-complexity bk-models based on PAX_1 and Horn metarules. For the many ExpTime-complete DL, secure view computation does not increase asymptotic complexity. So far, the best upper complexity bound for computing secure views in the description logic underlying OWL DL (i.e. $\mathcal{SROIQ}(\mathcal{D})$) is $\mathrm{coNP}^{N2ExpTime}$.

Finally, we have provided a prototype implementation of the low-complexity frameworks based on PAX_1 and Horn metarules using incremental engine versions available for Pellet and ELK to avoid repeated classifications in the iterative construction of $f_{\mathcal{M}}$. Metarule bodies are evaluated with SPARQL. Secure views are constructed off-line, so no overhead is placed on user queries, and this approach is expected to be applicable in practice.

Bibliography

[1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[2] F. Baader, M. Knechtel, and R. Peñaloza. A generic approach for large-scale ontological reasoning in the presence of access restrictions to the ontology's axioms. In *International Semantic Web Conference*, pages 49–64, 2009.

[3] J. Bao, G. Slutzki, and V. Honavar. Privacy-preserving reasoning on the semantic web. In *Web Intelligence*, pages 791–797. IEEE Computer Society, 2007.

[4] J. Biskup and P. A. Bonatti. Lying versus refusal for known potential secrets. *Data Knowl. Eng.*, 38(2):199–222, 2001.

[5] J. Biskup and P. A. Bonatti. Controlled query evaluation for enforcing confidentiality in complete information systems. *Int. J. Inf. Sec.*, 3(1):14–27, 2004.

[6] J. Biskup and P. A. Bonatti. Controlled query evaluation for known policies by combining lying and refusal. *Ann. Math. Artif. Intell.*, 40(1-2):37–62, 2004.

[7] P. A. Bonatti and L. Sauro. A confidentiality model for ontologies. In *International Semantic Web Conference (1)*, pages 17–32, 2013.

[8] W. Chen and H. Stuckenschmidt. A model-driven approach to enable access control for ontologies. In H. R. Hansen et al., editor, *Wirtschaftsinformatik*, volume 246 of *books@ocg.at*, pages 663–672. Österreichische Computer Gesellschaft, 2009.

[9] B. Cuenca Grau and I. Horrocks. Privacy-preserving query answering in logic-based information systems. In M. Ghallab, C. D. Spyropoulos, N. Fakotakis, and N. M. Avouris, editors, *ECAI*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 40–44. IOS Press, 2008.

[10] B. Cuenca Grau and B. Motik. Importing ontologies with hidden content. In B. Cuenca Grau et al., editor, *Description Logics*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[11] Eldora, M. Knechtel, and R. Peñaloza. Correcting access restrictions to a consequence more flexibly. In R. Rosati, S. Rudolph, and M. Zakharyaschev, editors, *Description Logics*, volume 745 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

[12] P. Hitzler and T. Lukasiewicz, editors. *Web Reasoning and Rule Systems - 4th Int. Conference, RR 2010.*, volume 6333 of *Lecture Notes in Computer Science*. Springer, 2010.

[13] M. Knechtel and H. Stuckenschmidt. Query-based access control for ontologies. In Hitzler and Lukasiewicz [12], pages 73–87.

[14] P. Stouppa and T. Studer. Data privacy for knowledge bases. In S. N. Artëmov and A. Nerode, editors, *LFCS*, volume 5407 of *LNCS*, pages 409–421. Springer, 2009.

[15] J. Tao, G. Slutzki, and V. Honavar. Secrecy-preserving query answering for instance checking in $\mathcal{EL}$. In Hitzler and Lukasiewicz [12], pages 195–203.