Optimized Construction of Secure Knowledge-Base Views

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

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

Abstract. In this paper we examine a confidentiality framework for constructing safe KB views with respect to the object-level and the meta-level background knowledge that users may exploit to reconstruct secrets. In particular, we will present a first implementation of our framework equipped with several optimization techniques that are assessed experimentally in a concrete e-health scenario.

1 Introduction

Recently, the Semantic Web has been increasingly used to encode sensible knowledge on individuals, companies and public organizations. As reasoning techniques make it possible to extract implicit information, any access control method that does not deal with inference fails to ensure privacy [1, 10].

The most popular security criterion is that the published view of a knowledge base should not entail any secret sentence [3, 9, 14]. However, such a model guarantees confidentiality just in the case the filtered knowledge base is the only source of information. On the contrary, various sources of background knowledge can be exploited to reconstruct secrets. Background knowledge can be object-level knowledge of the domain of interest, e.g. auxiliary ontologies, as well as meta knowledge about which kind of information the knowledge base is expected to represent. For instance, suppose a hospital allows to know whether a patient has been hospitalized but omits to reveal where, if she is in the infective disease ward. Since a hospital's *KB* is expected to have complete knowledge about which patients are in which ward, from the fact that John has been admitted to the hospital and yet he does not appear to be located in any ward, a user can reconstruct he is affected by some infection.[1]

To tackle the vulnerabilities arising from these scenarios, [7] has provided a fully generic formalization of object-level and meta-level background knowledge, a confidentiality model which neutralizes the inference-based attacks that exploit such knowledge, and – since the user's background knowledge is not directly possessed by the knowledge engineer – a rule-based methodology to safely approximate it.

As the works in [11, 12], our model is inspired by the literature on *Controlled Query Evaluation* ([4, 5, 6]). However, the two approaches differ in many aspects, including: *(i)* [11, 12] focus on conjunctive queries, while we focus on subsumption and instance

[1] For further details see the analogous Example 1 in [7]. In general, meta knowledge helps in preventing *attacks to complete knowledge* and *attacks to the signature*.

checking; *(ii)* in our framework secrets can be both intensional and extensional axioms, whereas in [11, 12] they can only be extensional facts; *(iii)* although [11, 12] can deal with object-level background knowledge, meta knowledge is not taken into account.

Regarding complexity issues, in [7] it has been shown that by using Horn rules to encode the user's meta knowledge, if the underlying DL is tractable, then the filtering secure function is tractable too.[2] Although such promising theoretical properties suggest that the framework can be practically used, they are still to be assessed experimentally. In this paper, we present SOVGen, a first prototype suited for a concrete e-health scenario. In particular, extensional data is encoded in realistic electronic health records conforming to the standard HL7 v.3 - CDA Rel.2. We approximate the user's background knowledge with the SNOMED-CT ontology, together with an ontology establishing the mapping between SNOMED-CT concepts and ICD-9CM codes that occur in the records. The user's meta knowledge, on the other hand, consists of *(i)* bridge metarules that permit to identify SNOMED-CT concepts starting from the specific encoding of the records required by CDA, as well as *(ii)* metarules that establish relationships between medications, diseases, medical procedures, etc.

Sec. 2 will provide a general overview on the theoretical model; due to space limitations we refer to [7] for technical proofs. In Sec. 3 we will describe the algorithm underlying SOVGen together with its optimizations. Sections 4 and 5 describe the experimental settings and performance analysis, respectively. Sec. 6 concludes the paper.

2 The Model

We assume the reader to be familiar with description logics, and refer to [2] for all definitions and results. We assume a fixed, denumerable signature Σ, specifying the names of *concepts*, *roles*, and *individuals*, and a reference logical language $\mathcal{L}$ is generated from Σ by the grammar of a DL. Unless stated otherwise, by *axioms* we mean members of $\mathcal{L}$; a *knowledge base* is any finite subset of $\mathcal{L}$. The notion of *logical consequence* is the classical one; for all $K \subseteq \mathcal{L}$, the logical consequences of K will be denoted by $Cn(K)$ ($K \subseteq Cn(K) \subseteq \mathcal{L}$).

Let KB be a knowledge base and $S \subseteq \mathcal{L}$ a set of secrecies. Generally speaking, the confidentiality of S is preserved if a user cannot expect to discover any secret by querying the system. A possible attempt to protect the secrets is to use a view KB' which is a maximal subset of KB that entails no secret, $Cn(KB') \cap S = \emptyset$.

Unfortunately, in case some background knowledge is available to the user, this mechanism could not ensure confidentiality. Frequently, part of the domain knowledge is not axiomatized in KB. In such cases a user can import some external ontology or RDF repository BK to infer more than she is allowed to. Moreover, she may possess some meta knowledge about KB. For instance, a hospital's KB is expected to have complete knowledge about its patients; a company's KB is likely to encode complete information about its employees, etc. Such meta knowledge can be represented epistemically as a set of possible knowledge bases PKB, queries can be then used to narrow PKB until the user is able to reconstruct a secret.

[2] Non-Horn metarules can be safely approximated with Horn metarules; the price to pay is a loss of *cooperativeness*, i.e. a reduction of the information available to the user.

Summarizing, we introduce a general confidentiality model which takes into account object-level and meta-level background knowledge.

Definition 1 ([7]). *A* bk-model *is a tuple* $\mathcal{M} = \langle KB, f, S, PKB, BK\rangle$ *where* KB *is a knowledge base,* $f : \wp(\mathcal{L}) \to \wp(\mathcal{L})$ *is a* filtering function *mapping each knowledge base* K *on a view* $f(K) \subseteq Cn(K)$*,* $S \subseteq \mathcal{L}$ *is a set of secrecies,* $BK \subseteq \mathcal{L}$ *is a set of axioms encoding the users' object-level knowledge,* $PKB \subseteq \wp(\mathcal{L})$ *is a set of possible knowledge bases encoding users' meta knowledge.*

The view of KB released to a user is $f(KB)$. Intuitively, f is secure if for each secret s there exists a possible knowledge base $K \in PKB$ such that *(i)* KB and K have the same observable behavior, that is, as far as the user knows, the knowledge base might be K, and *(ii)* K and the object-level background knowledge BK do not suffice to entail s.

Definition 2. *A filtering function* f *is* secure *(w.r.t.* $\mathcal{M}$*) iff for all* $s \in S$*, there exists* $K \in PKB$ *such that 1)* $f(K) = f(KB)$ *and 2)* $s \notin Cn(K \cup BK)$*.*

In the rest of the paper we focus on concrete scenarios where all the components of bk-models are finite. Moreover, we tacitly assume that no secret is violated a priori, that is, for all secrets $s \in S$ there exists $K \in PKB$ such that $s \notin Cn(K \cup BK)$.[3]

Clearly, Definition 2 just formalizes our desiderata, consequently the next step is to exhibit a *secure filtering function*. This function is formulated as an iterative process where for each axiom that, according to the user's meta knowledge, may possibly occur in the knowledge base a *censor* decides 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 all axioms occurring in at least one possible knowledge base, i.e. $PAX = \bigcup_{K' \in PKB} K'$. Let $\nu = |PAX|$ and $\alpha_1, \ldots, \alpha_i, \ldots, \alpha_\nu$ be any enumeration of PAX. 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 $K^+ = \bigcup_{i\leq \nu} K_i^+$, $K^- = \bigcup_{i\leq \nu} K_i^-$, and $f_{\mathcal{M}}(K) = K^+$. The iterative 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^-).

In order to define the censor we need an auxiliary definition that captures all the consequences of the background knowledge BK and the meta knowledge PKB refined by a constraint $\langle X^+, X^-\rangle$. Let $Cn_{\mathcal{M}}(X^+, X^-)$ be the set of all axioms $\alpha \in \mathcal{L}$ such that

$$\text{for all } K' \in PKB \text{ such that } K' \text{ satisfies } \langle X^+, X^-\rangle, \alpha \in Cn(K' \cup BK)\,. \tag{1}$$

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

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

$$censor_{\mathcal{M}}(X^+, X^-, \alpha) = \begin{cases} true & \text{if there exists } s \in S \text{ s.t. either } s \in Cn_{\mathcal{M}}(X^+ \cup \{\alpha\}, X^-) \\ & \text{or } s \in Cn_{\mathcal{M}}(X^+, X^- \cup \{\alpha\}); \\ false & \text{otherwise.} \end{cases} \quad (2)$$

In other words, the censor checks whether telling either that α is derivable or not to a user – aware that the knowledge base satisfies $\langle X^+, X^- \rangle$ – restricts the set of possible knowledge bases enough to conclude that a secret s is entailed by the knowledge base enriched with 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 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). This way, roughly speaking, the knowledge bases that entail s are given the same observable behavior as those that don't. Thm 1 in [7] shows that $f_{\mathcal{M}}$ is secure w.r.t. $\mathcal{M}$.

Remark 1. Observe that our method is inspired by CQE based on lies and/or refusals ([4, 5, 6] etc). Technically we use *lies*, because rejected queries are not explicitly marked. 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), $f_{\mathcal{M}}(KB)$ 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, nor meta knowledge.

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

Definition 3. *Let $\mathcal{M} = \langle KB, f, S, PKB, BK \rangle$ and $\mathcal{M}' = \langle KB', f', S', PKB', BK' \rangle$ be two bk-models, we write $\mathcal{M} \leq_k \mathcal{M}'$ iff $KB = KB'$, $f = f'$, $S = S'$, $PKB \supseteq PKB'$ and $BK \subseteq BK'$.*

Then, it is easy to see that $\mathcal{M}'$ is a safe approximation of $\mathcal{M}$, that is if f is secure w.r.t. $\mathcal{M}'$, then it is also secure w.r.t. $\mathcal{M}$ (Proposition 2, [7]).
Consequently, a generic advice for estimating BK consists in *(i)* including public ontologies and triple stores formalizing relevant knowledge and *(ii)* modeling as completely as possible the integrity constraints satisfied by the data, as well as role domain and range restrictions and disjointness constraints.

While BK can be represented with standard languages (e.g. OWL, RDF, etc.), user's meta knowledge requires an ad-hoc language for defining PKB. Here we express PKB as the set of all theories that are contained in a given set of *possible axioms* PAX and satisfy a finite set MR of *metarules* like:

$$\alpha_1, \ldots, \alpha_n \Rightarrow \beta_1 \mid \ldots \mid \beta_m \quad (n \geq 0, m \geq 0)\,, \quad (3)$$

where all α_i and β_j are in $\mathcal{L}$ ($1 \leq i \leq n$, $1 \leq j \leq m$). For all metarules r, let $body(r) = \{\alpha_1, \ldots, \alpha_n\}$ and $head(r) = \{\beta_1, \ldots, \beta_m\}$.

Informally, (3) 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 $ground_K(MR)$ be the ground instantiation of MR where metavariables are uniformly replaced by the individual constants occurring in K in all possible ways.

A set of axioms $K \subseteq \mathcal{L}$ *satisfies* a ground metarule r if either $body(r) \nsubseteq Cn(K)$ or $head(r) \cap Cn(K) \neq \emptyset$. In this case we write $K \models_m r$. Moreover, if K satisfies all the metarules in $ground_K(MR)$ then we write $K \models_m MR$. Therefore the formal definition of PKB now becomes:

$$PKB = \{K \mid K \subseteq PAX \wedge K \models_m MR\}. \tag{4}$$

In this paper, we assume that MR consists of Horn metarules ($|head(r)| \leq 1$) and $PAX = KB \cup \bigcup_{r \in ground_{KB}(MR)} head(r)$. Under such hypothesis, it can be shown that if all the axioms in KB, PKB, BK, and S belong to a tractable DL, and the number of distinct variables in MR is bounded by a constant, then $f_{\mathcal{M}}$ can be calculated in polynomial time.

3 Implementation overview

In this section we introduce SOVGen, the prototypical implementation of the confidentiality model illustrated in Section 2 based on Horn metarules. By standard logic programming techniques, a minimal $K \subseteq PAX$ satisfying the set of metarules and the constraints K^+ can be obtained with the following polynomial construction:

$$K_0 = K^+, \quad K_{i+1} = K_i \cup \bigcup \{\, head(r) \mid r \in ground_{K_i}(MR) \wedge body(r) \subseteq Cn(K_i) \,\}$$

It can be proved that the sequence limit $K_{|PAX|}$ satisfies $\langle K^+, K^- \rangle$ as well if $K_{|PAX|}$ does not entail an axiom in K^-. Then, for all $s \in S$, s activates the censor iff s is a consequence of $K_{|PAX|} \cup BK$. For further details refer to [7].

Algorithm 1 represents the abstract algorithm underlying SOVGen. The sets M_M and M_G constitute a partition of MR based on the metarules' type (ground or containing metavariables). Iterating over the axioms $\alpha \in PAX$ (*lines 6-25*), at each step K collects all the axioms of PAX that does not contribute to the entailment of secrets. The repeat-until loop (*lines 9-17*) computes the deductive closure K' of K under the set of metarules MR. In particular, for each ground metarule (*lines 10-13*) we evaluate a conjunctive query (encoded in *line 11*) in order to check if m body is satisfied by the current K'. Similarly, for each metarule containing metavariables (*lines 14-16*), we obtain all possible bindings for the metavariables in the body of m by means of a conjunctive query evaluation (*line 15*). The sequence of steps described above is iterated until a fixpoint is reached (*line 17*). At this point the condition $Cn(K') \cap K^- \models \emptyset$ is verified (*line 18*). It is now possible to determine the value of the censor for α. We first check that no secret is entailed from the minimal K (*line 19*) enreached with BK. Finally, we can safely include α in the view only if it is entailed by KB (*line 21*). Otherwise, the set K^-

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

is updated (*line 25*). Note that, 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 (*lines 13, 23*).

A careful analysis of the algorithm immediately points out: *(1)* the opportunity to apply a process of modularization designed to reduce the size of very large background knowledge bases (such as SNOMED-CT). In fact, many of the axioms in a large BK are reasonably expected to be irrelevant to the given view; *(2)* the need of techniques for effective conjunctive query evaluation.[4]

With respect to point (1), we investigate the use of *module extractors* [17, 16] on the background knowledge bases in order to make reasoning focus on relevant knowledge only. Experimental results show that the modules extracted are on average two

[4] 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.

or three orders of magnitude smaller than the initial BKs which drastically improves performance.

With respect to point (2), the presence of technologies that permit native conjunctive query evaluation reveals fundamental to achieve efficient framework implementation. Nowadays SPARQL[5], constitute a de facto standard when it comes to conjunctive query answering. It has been recently extended with the OWL Direct Semantics Entailment Regime in order to permit reasoning over OWL ontologies. Unfortunately, only few tools provide support to this new semantics. Among those our choice fell on *Apache Jena Semantic Web Toolkit*[6] (for more information and motivations see [8]). A valid alternative to the consolidated SPARQL engines proves to be *OWL-BGP*[7], a relatively new framework for parsing SPARQL basic graph patterns (BGPs) to OWL object representation and their assessment under the OWL Direct Semantics Entailment Regime. *OWL-BGP* incorporates various optimization techniques [15] including query rewriting and a cost-based model[8] for determining the order in which conjunctive query atoms are evaluated. As we will see in Section 5 the performance of the query evaluation module of SOVGen is unacceptable when Jena is used and not quite satisfactory when *OWL-BGP* is adopted[9]. As an alternative to the above frameworks for conjunctive query evaluation we propose an hoc module, called *Metarule Evaluation Engine (MEE)*, that aims to take advantage of the specific nature of the Horn metarules and incremental reasoning techniques of ELK [13].

Metarule Evaluation Engine (MEE). The evaluation algorithm is based on direct calls to an incremental reasoner. In the following we provide a brief description of the procedure employed for the evaluation of the different types of metarules.

The evaluation of a ground metarule r requires checking that all the axioms $\alpha_1, \ldots, \alpha_n$ in $body(r)$ are entailed by $K^{'}$. The algorithm takes advantage of *short circuit evaluation* techniques that permit to end the evaluation as soon as $K^{'} \not\models \alpha_i$ and memoization of the atoms α_i satisfied in previous iterations in order to avoid their re-evaluation.

The evaluation of metarules with metavariables, on the other hand, comprises a preprocession step that partition the atoms $\alpha_1, \ldots, \alpha_n$ in the metarule body in sets of *connected components*. Within a component, atoms (that in this case can be viewed as axiom templates) share common metavariables, while there are no metavariables shared between atoms belonging to different *connected components*. Evaluating together templates belonging to non-related components increases unnecessarily the amount of intermediate results, whereas it is sufficient to combine the results for the single components. Furthermore, for some types of templates, such as $C(X)$, it is possible to retrieve the solutions directly from the reasoner, instead of verifying the satisfiability of each compatible mapping for the metavariable X. Although this can trigger some internal controls, most of the methods of reasoners are highly optimized. Other more complex

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

[6] http://jena.apache.org/

[7] https://code.google.com/p/owl-bgp/

[8] The cost calculation is based on information about instances of concepts and roles extrapolated from an abstract model built by reasoners that implement Tableaux reasoning algorithms.

[9] Note that evaluation of ground metarules results in *SPARQL ASK* query (line11 of Alg.1), while evaluation of metarules with metavariables in *SPARQL SELECT* query (line15 of Alg.1).

templates, like the property assertions $R(X, Y)$, do not allow the evaluation via dedicated reasoning tasks and require satisfiability check for each possible instantiation. Consequently, within each connected component, the evaluation is performed considering first all atoms of the type $C(X)$ for the purpose of restricting as much as possible the compatible mappings for the metavariables, then the atoms $R(X, Y)$ (X or Y may possibly be an individual constant) are considered.

Note that, unlike the previous engines, MEE does not need to initialize the inference model on each step of the repeat-until loop. In fact, the queries are evaluated through a number of calls to the ELK reasoner, that make it possible to exploit the characteristics of incremental classification.

4 Experimental Settings

In this section we present synthetic test cases which have been specifically designed to simulate the employment of SOVGen in a e-health scenario. In particular, each test case represents the encoding of sensitive data in a CDA-compliant electronic health record.[10]

According to the theoretical framework each test case comprises four different components: the ontology *KB* that contains confidential data to be protected; an ontology *MR* encoding the user meta knowledge with a set of metarules; a set S of secrets; a series of ontologies representing the user's object-level background knowledge *BK*.

KB generation. KB is generated as a set of assertions instantiating the PS ontology. PS encodes a patient summary clinical document following the HL7 Implementation Guide for CDA Rel.2 Level 3: Patient Summary. As it can be seen in Figure 1, PS currently provide a support for encoding information about (i) history of assumed medications; (ii) clinical problem list including diagnosis, diagnostic hypothesis and clinical findings; (iii) history of a family member disease; (iv) list of the procedures the patient has undergone; (v) list of relevant diagnostic tests and laboratory data. Note that, according to the CDA standards a disease in the PS ontology is represented by a ICD-9CM code, while pharmaceutical products and procedures are represented by a SNOMED CT codes. For example, <code code="64572001" codeSystemName="SNOMED CT"/> stands for an instance of the SNOMED CT concept Disease (SCT_64572001). The type of sections to be generated are randomly chosen among those mentioned above. A disease (resp. product, procedure, test) code to associate to the entries is chosen as a random leaf of the corresponding Disease (resp. Pharmaceutical/biologic product, Procedure by site, Measurement procedure, Imaging) concept of the SNOMED CT ontology. In case a disease code is needed, the ICD-9CM code corresponding to the SNOMED CT one is retrieved and the equivalence is added to a background knowledge ontology named EQIV-RL.

Metarule generation. The knowledge encoded in KB gives rise to several possible types of metarules. Bridge metarules associate a ICD-9CM/SNOMED CT code to the concept in the respective ontology. For instance,

$$\mathrm{CD}(C),\ \mathrm{dtpCode}(C, 64572001),\ \mathrm{dtpCodeSystem}(C, \mathrm{SNOMED\text{-}CT}) \Rightarrow \mathrm{SCT\_64572001}(C)$$

[10] Clinical Document Architecture (CDA) is a standard for information exchange, based on the Health Level 7 Reference Information Model.

makes it possible to derive that a code instance C is in fact an instance of the Disease concept in SNOMED CT.

The second type of metarules concerns the pharmaceutical products. The presence of a drug in the history of medication use implies that the patient suffers (certainly or with a great probability) from a specific pathology or has undertaken a specific procedure. Consider the following example of metarule which says that the presence of a medicine with active ingredient Phenytoin (SCT_40556005) indicates that the patient suffers from some kind of Epilepsy (SCT_84757006):

$$\begin{aligned}&\text{Patient}(P),\ \text{SubstanceAdministration}(SA),\ \text{Consumable}(C),\ \text{hasConsumable}(SA, C),\\ &\text{ManufacturedProduct}(MP),\ \text{hasManufacturedProduct}(C, MP),\ \text{Material}(M),\\ &\text{hasManufacturedMaterial}(MP, M),\ \text{SCT\_40556005}(CD),\ \text{hasCode}(M, CD)\\ &\Rightarrow \exists \text{suffer.SCT\_84757006}(P)\end{aligned}$$

The third type of metarules concerns the problems section. In particular the presence of a diagnosis (resp. diagnostic hypothesis) indicates that the patient suffer (resp. possibly suffer) a certain pathology.

Other types of metarules apply to the family history – e.g. a patient could be subject to a family members' disease – and the procedures section. For instance, the metarule

$$\text{Patient}(P),\ \text{Procedure}(I),\ \text{SCT\_77465005}(C),\ \text{hasCode}(I, C) \Rightarrow \text{subject}(P, C)$$

allows to entail that the presence of an organ transplantation (SCT_77465005) in the procedure section indicates that the patient is subject to transplantation.

Note that the generation of MR is not completely random for a part of the metarules. In order to obtain a nontrivial reasoning, during the KB generation, together with the creation of a section' entry is also created one or more corresponding bridge metarules and a metarule corresponding to the section in question. A second part of metarules are constructed by randomly selecting appropriate SNOMED CT concepts as needed. The adoption of such approach guarantees that al least part of metarules are actually fired during the secure ontology view generation. Furthermore, observe that the there are actually two levels of metarules, the bridge metarules constitute a precondition for the activation of the others.

Secrets generation. The ontology S is randomly generated as a set of assertions of the types:

$$\exists \text{suffer}.X(p) \quad \exists \text{possiblySuffer}.X(p) \quad \exists \text{possibleSubject}.X(p) \quad \exists \text{subject}.Y(p)$$

where X (resp. Y) is chosen as a random subconcept of the Disease (resp. Procedure) concept of the SNOMED CT ontology.

Background knowledge. The background knowledge *BK* is approximated by means of the PS, SNOMED-CT and the previously mentioned EQIV-RL ontologies.

5 Performance Analysis

In this section we present a performance analysis of SOVGen. Scalability evaluations have been carried out on synthetic test cases as described in Section 4. The size of *KB* is given by the parameter *KB-size* as the number of assertions occurring in the ontology.

```
<clinicalDocument>
   <recordTarget>
      <patientRole>
         <patient> . . . </patient>
      </patientRole>
   </recordTarget>
   <structuredBody>
      <section> <code code='10160-0' codeSystemName='LOINC'/> <!-- HISTORY OF MEDICATION USE -->
         <entry> . . . </entry>
      </section>
      <section> <code code='11450-4' codeSystemName='LOINC'/> <!-- CLINICAL PROBLEM LIST -->
         <entry> . . . </entry>
      </section>
      <section> <code code='10157-6' codeSystemName="LOINC"/> <!-- FAMILY MEMBER DISEASES -->
         <entry> . . . </entry>
      </section>
      <section> <code code='47519-4' codeSystemName='LOINC'/> <!-- HISTORY OF PROCEDURES -->
         <entry> . . . </entry>
      </section>
      <section> <code code='30954-2' codeSystemName="LOINC"/> <!-- RELEVANT DIAGNOSTIC TESTS -->
         <entry> . . . </entry>
      </section>
   </structuredBody>
</clinicalDocument>
```

Fig. 1. HL7 CDA Rel.2 Patient Summary

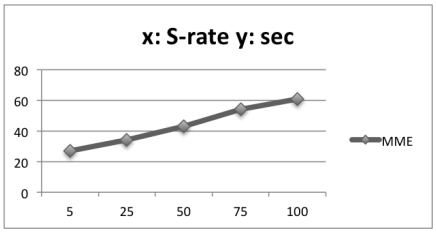
Then, the size of MR, *MR-rate*, is the ratio between the number of metarules and the number of assertions in *KB*. Finally, the size of S is determined by the parameter *S-rate* that specifies the ratio $|S|/|KB|$.

The experiments were performed on an Intel Core i7 2,5GHz laptop with 16GB and OS X 10.10.1, using Java 1.7 configured with 8GB RAM and 4GB stack space. Each reported value is the average execution time of five runs over five different ontologies. Note that given the amount of background knowledge (consider that SNOMED-CT describes about 300K concepts) the use of module extraction techniques improves the computation time of two–three orders of magnitude at a cost of about 30 sec of overhead.

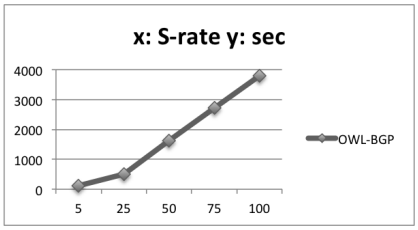
In Figure 2, the left (resp. right) column shows the experimental results obtained by using MEE (resp. OWL-BGP) to evaluate metarules – no result for Jena is reported as the execution time on all the test cases exceeded 1 hour time-out. Figures 2(a) and 2(b) report the execution time as the amount of secrets grows. Both *MR-rate* and *KB-size* are fixed, respectively to 10% and 200 assertions. Note that, MEE outperforms OWL-BGP of 1–2 orders of magnitude. Figures 2(c) and 2(d) show the impact of MR-rate when *KB-size* is fixed to 200 and *S-size* to 10%. Here, MEE runs about 10 times faster than OWL-BGP. Finally, Figures 2(e) and 2(f) illustrate the way the execution time changes as the the size of *KB* increases. Again MEE is 10^2 faster than OWL-BGP.

6 Conclusions

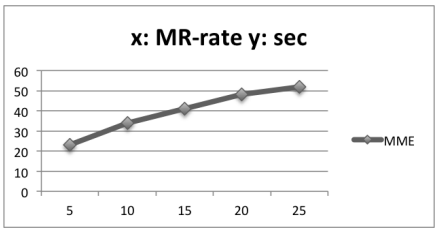
In [7] a novel confidentiality model has been introduced which adapts Controlled Query Evaluation to the context of Description Logics, and extends it by taking into account object-level and meta background knowledge. Here, we have presented SOVGen, a first implementation of this methodology that has been specialized to deal with a concrete



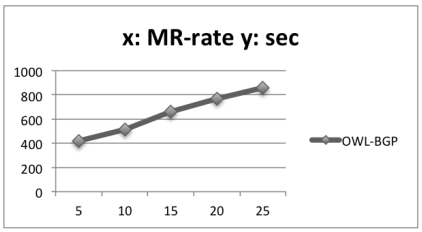
(a) KB-size=200, MR-rate=10%



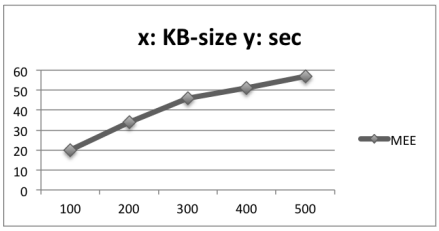
(b) KB-size=200, MR-rate=10%



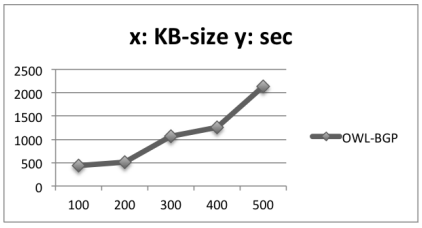
(c) KB-size=200, S-rate=10%



(d) KB-size=200, S-rate=10%



(e) MR-rate=10%, S-rate=25%



(f) MR-rate=10%, S-rate=25%

Fig. 2. Secure view construction time with MEE e OWL-BGP on variation of the parameters S-rate, MR-rate and KB-rate

e-health application. In order to maximize performance, we have compared different reasoning tools and designed several optimization techniques. Then, we assessed SOVGen experimentally by using realistic electronic health records that refer to SNOMED-CT concepts, and Horn rules to represent meta knowledge. In particular, we observed that module extraction techniques and a suitable, ad-hoc metarule evaluation engine – which intensively exploit ELK incremental reasoning – largely outperform general conjunctive query evaluation engines.

Considering that secure views are constructed off-line – so that no overhead is placed on user queries – performance analysis shows that SOVGen is close to meet practical use in this application scenario. In future work, we aim at improving the system with new optimizations, and extending it to general rules.

Bibliography

[1] F. Abel, J. L. D. Coi, N. Henze, A. W. Koesling, D. Krause, and D. Olmedilla. Enabling advanced and context-dependent access control in RDF stores. In K. Aberer et al., editor, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, volume 4825 of *LNCS*, pages 1–14. Springer, 2007.

[2] 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.

[3] 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.

[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 H. Alani, L. Kagal, A. Fokoue, P. T. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. F. Noy, C. Welty, and K. Janowicz, editors, *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, volume 8218 of *Lecture Notes in Computer Science*, pages 17–32. Springer, 2013.

[8] P. A. Bonatti, L. Sauro, and I. Petrova. A mechanism for ontology confidentiality. In L. Giordano, V. Gliozzi, and G. L. Pozzato, editors, *Proceedings of the 29th Italian Conference on Computational Logic, Torino, Italy, June 16-18, 2014.*, volume 1195 of *CEUR Workshop Proceedings*, pages 147–161. CEUR-WS.org, 2014.

[9] 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.

[10] G. Flouris, I. Fundulaki, M. Michou, and G. Antoniou. Controlling access to RDF graphs. In A.-J. Berre, A. Gómez-Pérez, K. Tutschku, and D. Fensel, editors, *FIS*, volume 6369 of *Lecture Notes in Computer Science*, pages 107–117. Springer, 2010.

[11] B. C. Grau, E. Kharlamov, E. V. Kostylev, and D. Zheleznyakov. Controlled query evaluation over OWL 2 RL ontologies. In H. Alani, L. Kagal, A. Fokoue, P. T. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. F. Noy, C. Welty, and K. Janowicz, editors, *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, volume 8218 of *Lecture Notes in Computer Science*, pages 49–65. Springer, 2013.

[12] B. C. Grau, E. Kharlamov, E. V. Kostylev, and D. Zheleznyakov. Controlled query evaluation over lightweight ontologies. In M. Bienvenu, M. Ortiz, R. Rosati, and M. Simkus, editors, *Informal Proceedings of the 27th International Workshop on Description Logics, Vienna, Austria, July 17-20, 2014.*, volume 1193 of *CEUR Workshop Proceedings*, pages 141–152. CEUR-WS.org, 2014.

[13] Y. Kazakov, M. Krötzsch, and F. Simancik. The incredible ELK - from polynomial procedures to efficient reasoning with el ontologies. *J. Autom. Reasoning*, 53(1):1–61, 2014.

[14] M. Knechtel and H. Stuckenschmidt. Query-based access control for ontologies. In P. Hitzler and T. Lukasiewicz, editors, *RR*, volume 6333 of *Lecture Notes in Computer Science*, pages 73–87. Springer, 2010.

[15] I. Kollia and B. Glimm. Optimizing SPARQL query answering over OWL ontologies. *CoRR*, abs/1402.0576, 2014.

[16] F. Martin-Recuerda and D. Walther. Axiom dependency hypergraphs for fast modularisation and atomic decomposition. In M. Bienvenu, M. Ortiz, R. Rosati, and M. Simkus, editors, *Proceedings of the 27th International Workshop on Description Logics (DL'14)*, volume 1193 of *CEUR Workshop Proceedings*, pages 299–310, 2014.

[17] U. Sattler, T. Schneider, and M. Zakharyaschev. Which kind of module should I extract? In B. Cuenca Grau et al., editor, *Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.