Undefined 1 (2009) 1–5
IOS Press

Privacy in Ontology-based Information Systems: A Pending Matter

Editor(s): Krzysztof Janowicz, Pennsylvania State University, USA and Pascal Hitzler, Wright State University, USA
Solicited review(s): Michel Dumontier, Carleton University, Canada; Paulo Pinheiro da Silva, The University of Texas at El Paso, USA
Open review(s): None

Bernardo Cuenca Grau *,
Oxford University Computing Laboratory
Oxford, United Kingdom
E-mail: berg@comlab.ox.ac.uk

Abstract. OWL ontologies are extensively used in the clinical sciences, with ontologies such as SNOMED CT being a component of the health information systems of several countries. Preserving privacy of information in ontology-based systems (e.g., preventing unauthorised access to system's data and ontological knowledge) is a critical requirement, especially when the system is accessed by numerous users with different privileges and is distributed across applications. Unauthorised disclosure, for example, of medical information from SNOMED-based systems could be disastrous for government organisations, companies and, most importantly, for the patients themselves. It is to be expected that privacy-related issues will become increasingly important as ontology-based technologies are integrated in mainstream applications. In this short paper, I discuss several challenges and open problems, and sketch possible research directions.

Keywords: Ontologies, Semantic Web, Data Privacy

1. Background

Ontologies expressed in the Web Ontology Language (OWL) or its revision OWL 2 are already being used for applications in fields as diverse as biomedicine, astronomy and defence. For example, OWL ontologies are extensively used in the clinical sciences, with ontologies such as SNOMED CT being a component of health information systems of several countries.

OWL ontologies can be used to formally describe the meaning of data (e.g., electronic patient records in the case of a medical application). Applications can then exploit ontologies to process the associated data in a more intelligent way. For example, a medical ontology describing patient record data may contain information such as "every patient with a mental disorder must be treated by a psychiatrist", "schizophrenia is a kind of psychosis", and "psychosis is a kind of mental disorder"; if John's medical record states that he suffers from schizophrenia, then an ontology can be used to conclude that he suffers from a mental disorder and must be treated by a psychiatrist.

Preserving privacy of the information in ontology-based systems (e.g., preventing unauthorised access to system's data and ontological knowledge) is a critical requirement, especially when the system is accessed by numerous users with different privileges and is distributed across applications. In particular, there may be multiple groups of users who want to access and retrieve information from the same ontology and its associated data sources. In this setting, different access rights may be granted to each of these groups and privacy preservation implies, for example, ensuring that users can only retrieve (either directly or indirectly via logical inference) the information they are allowed to access. The unauthorised disclosure, for ex-

*The author is supported by a Royal Society University Research Fellowship.

ample, of medical information from SNOMED-based systems (e.g., the identity of schizophrenic patients in a hospital) could be disastrous for government organisations, companies and, most importantly, for the patients themselves.

Data privacy in information systems is a long standing research area, which is particularly active in databases (DBs) (e.g., [3,23,4,10,20,5]). Very little is known, however, about privacy in the context of OWL ontologies and only recently has research been conducted in this direction [24,2,7,8].

Existing work on data privacy in databases focuses mainly on *complete* relational DBs [20,19,10,4]. Ontologies, however, are strongly related to *incomplete* DBs [18,22], with the difference that ontology languages are typically much more expressive than DB schema languages.

In contrast to complete DBs, query evaluation requires taking into account *all* models of the incomplete DB (or ontology) to compute the *certain answers* to the query formula (that is, the answers logically inferred by the union of the schema/ontology and the data). In our previous example, the fact that John suffers from a mental disorder and must be treated by a psychiatrist is not explicitly given; however, it can be deduced as a consequence of given information. These inferences may involve non-obvious interactions between different pieces of information in the system.

Data privacy in the context of incomplete or semi-structured DBs has only recently been investigated [5,12]. Furthermore, these works do not consider the presence of complex dependencies such as the ones present in OWL ontologies.

It is to be expected that privacy-related issues will become increasingly important as ontology-based technologies are integrated in mainstream applications. In the remainder of this paper, I discuss several challenges and open problems, and sketch possible research directions.

2. General Challenges

In my discussion, I will focus on two general challenges for future research. The first one is related to the *design* of a privacy-preserving ontology-based system, whereas the second one concerns the *(re)use* of such system by external applications.

To illustrate the first challenge, consider the information system of a hospital whose privacy policy should prevent Bob from accessing the relationship between patients and their medical conditions. In DBs, access control has traditionally been achieved by presenting users with (relational) *views* that omit the sensitive information (e.g., the table relating patients to medical conditions) [1,14]. In the case of ontologies, however, providing a view that filters out such explicit statements may not be sufficient to ensure privacy.

Suppose that Bob knows that John has only been in the hospital once and, on that occasion, he was treated by both Dr. Smith (a gastroenterologist) and Dr. Andrews (a psychiatrist); from the ontology Bob knows that gastroenterologists only treat gastric diseases and psychiatrists only treat either mental disorders or psychosomatic illnesses; moreover, a disease cannot be both a mental disorder and a gastric disease and, if a disease is both psychosomatic and gastric, it must be a form of irritable bowel syndrome. Bob could then infer that John suffers from a kind of irritable bowel syndrome. Thus, restricted information can be leaked via logical inference.

Therefore, the first challenge is the development of the theoretical foundations and practical techniques necessary for the *design* of systems that provide provable privacy guarantees as well as to gain an understanding of the limitations of these guarantees.

The second challenge follows from the previous discussion, which suggests that access to information in an ontology-based system providing privacy guarantees should be *restricted* (i.e., the system's ontology and data cannot be published, at least not entirely). In the case of our previous example, to comply with the privacy requirements the system should not make public any information that would lead an external user to infer that John suffers from a kind of irritable bowel syndrome.

The system's owners may be reluctant to even publish the non-confidential information; for example, they may not be willing to distribute the contents of the ontology (even if data access is restricted), as doing so might allow competitors to plagiarise it; also, they might want to impose different costs for reusing parts of the ontology. This is the case with SNOMED CT, which is only available under a license agreement.

Currently, the only way for ontology-based applications to (re)use other ontologies and data sources is by means of OWL's *importing* mechanism [15]. OWL tools deal with imports by internally merging (i.e., constructing the union of the contents of) the relevant ontologies and the relevant data sources; hence the use of OWL's importing mechanism requires physical access to the entire contents of a system. If these

contents are not available due to access limitations, the use of OWL's importing mechanism is clearly no longer possible. As a consequence, further research is needed in order enable the effective (re)use of a privacy-preserving system by external applications.

Therefore, the second challenge is to investigate the conditions under which an application can effectively *(re)use* an ontology-based system to which access limitations have been imposed due to privacy considerations.

3. Design of a Privacy-preserving System

In this section, I argue that the design of a privacy-preserving ontology-based system requires addressing the following issues:

1. *Policy representation*: How can system designers establish in a declarative way what information should be inaccessible to which users?
2. *Models of interaction*: What kinds of queries can users pose to the system?
3. *Formalisation of users' prior knowledge*: How could system designers take into account the knowledge that users may already have acquired when querying the system (e.g., the results obtained from previous queries) and which could be used to violate the policies?
4. *Notions of policy violation*: What does it mean for users to discover, by interacting with the system and using their prior knowledge, information that is confidential according to the policy applied to them?

A privacy policy specifies, in a declarative way, which information should not be accessible to which users (or group of users defined, for example, according to a role-based access model) [4]. An important issue is to establish the way in which policies are to be represented by the designer of the system.

In the database theory literature, policies are often represented using various types of data-centric *queries* (e.g., conjunctive queries) [11,19,20]. The representation of policies as (conjunctive) queries has been recently proposed by [25] in the context of ontologies. In the case of ontology-based systems, however, schema information plays a key role and hence policy languages should also take into account what schema information should be visible to a given user and hence typical data-oriented database queries may not suffice to specify suitable policies.

In the context of Web services, the languages WS-Policy [26] and XACML [21] have been used to specify policies. These languages provide sophisticated features that could also be useful for ontology-based systems. They are, however, not equipped with a logic-based semantics. In fact, although there have been attempts to formalise them (e.g., [6,27]), it is not clear how policies in these languages should be interpreted and evaluated w.r.t. the system's ontology. Therefore, the following questions can be an interesting starting point for future research:

- What policy languages are suitable in the context of ontology-based systems?
- How do such languages relate to those used in the context of databases and Web services?

The representation of complex policies leads to the problem of designing and maintaining them; that is, policy designers may have difficulties understanding the consequences of their policies as well as detecting errors. For example, a policy P (applied to managers) is more general than P' (applied to employees) if all the access restrictions in P also apply to employees. It would be useful to automatically check whether this is so if the system's ontology is taken into account. Therefore, an interesting research direction is to investigate reasoning problems for assisting system designers in writing high-quality policies. Preliminary results in this direction have been reported in [16].

Once the relevant policies have been designed, the next problem is to formally specify what it means for users to *violate* the policy assigned to them (i.e., to find out, by interacting with the system, information that is confidential according to the relevant policy). To this end, a first step is to formally describe the *interaction between users and the system*. It is reasonable to assume that users interact with the system by submitting queries in a given query language. Depending on the policy P assigned to each user, the system then decides whether to answer or reject the user's query Q and, in the former case, which answers to provide (for example, the true complete answer, an incomplete answer, or even an incorrect answer!).

In order for the system to make informed decisions, the *user's prior knowledge* (e.g., the answers to users' previous queries) should be considered. Formalising such prior knowledge and its provenance can be extremely difficult because information may come from many sources and/or from interactions between different users and so assumptions need to be made. In our example, Bob could query the system and learn that

"John is treated by Dr. Andrews", or "Irritable bowel syndromes are gastric diseases"; also, he may access other systems with overlapping information (e.g., the NHS website saying that "Dr. Andrews is a psychiatrist"). The formalisation of policy representation, user-system interaction and user's prior knowledge leads to the question of how to formalise the problem of policy verification, which can be informally described as follows:

> *Policy verification*: Given users prior knowledge and the corresponding policy P, does answering a given user's query Q violate the policy?

The notion of *policy violation* is open to many interpretations, and an interesting research problem is to investigate suitable semantics explaining what it is meant by violating a policy in this setting.

Once policy verification has been formalised, it remains to be seen how and when policies are verified by the system. Two scenarios are particularly worth investigating:

- *Online auditing*, where the system decides "on the fly" to answer or to reject users' queries.
- *Offline auditing*, where an auditor checks "a-posteriori" whether the answers given to a user might have compromised the policy.

In the former case, it seems reasonable to assume that users have only access to the system itself, and hence the only sources of relevant prior knowledge are the results of their previous queries. Indeed, in an online scenario it is virtually impossible to find out what other sources of information a user may have had access to or which users might have exchanged information. In the latter case, however, an auditor conducts an investigation which may reveal, for example, that Bob has had access to certain information in other systems, or has exchanged certain information with another user; the auditor then tries to determine whether Bob could be blamed for a particular privacy breach.

4. External Use of a Privacy-preserving System

Our second challenge was to study the situation where an external ontology-based application A wants to (re)use an ontology-based system S whose content is not available due to privacy considerations. The goal is to allow users of A to formulate queries and obtain the corresponding answers with respect to *the union of* the contents of *both* A and S, but taking into account that access limitations have been imposed to S. A central issue is how to model such access limitations, and only recently there has been research in this direction.

The authors of [7] have proposed to use data-centric *views* to formalise such access limitations. Views are represented as conjunctive queries, are given a priori, and must be compliant with the relevant policies. View extensions are computed as certain answers w.r.t. the ontology and data in S. The system makes sure that information from S not implied by the views remains hidden.

The authors of [17] have studied the situation in which the designers of S "hide" a subset of the vocabulary of S by publishing a so-called *uniform interpolant*. The interpolant can be seen as a "reusable projection" of the system's ontology and data that contains no "hidden" symbols that coincides with S on all logical consequences formed using the remaining "visible" symbols [17].

In our recent work, we have proposed an approach in which access limitations are imposed by making S accessible only via a limited query interface that we call an *oracle* [9,13]. The oracle can answer only a class of "allowed" queries over S. Under certain assumptions, a so-called *import-by-query* algorithm can reason over the union of the contents of A and S without having physical access to the content of S, by only posing queries to the oracle for S. In this situation, users may not even be aware of the existence of the privacy-preserving system.

The results in [7,17,9,13] have opened new areas of research. However, they have also left many open problems and further research is needed before they can be incorporated in practical systems.

5. Conclusion

In this short paper, I have discussed recent research on privacy-related issues in the context of ontology-based information systems.

I have identified several challenges and open problems for future research, and have sketched possible research directions. Many interesting related topics have been left out, which I believe will be (or continue to be) active areas of research within the next few years. These include, among others, privacy in the context of RDF data, issues related to trust and data provenance, and data and ontology anonymisation, among others.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

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

[3] E. Bertino and R. S. Sandhu. Database security-concepts, approaches, and challenges. *IEEE Trans. Dependable Sec. Comput.*, 2(1):2–19, 2005.

[4] 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.

[5] J. Biskup and T. Weibert. Keeping secrets in incomplete databases. *Int. J. Inf. Sec.*, 7(3):199–217, 2008.

[6] J. Bryans. Reasoning about XACML policies using CSP. In *Proc. of SWS*, 2005.

[7] D. Calvanese, G. D. Giacomo, M. Lenzerini, and R. Rosati. View-based query answering over description logic ontologies. In *Proc. of KR-2008*. AAAI Press, 2008.

[8] B. Cuenca Grau and I. Horrocks. Privacy-preserving query answering in logic-based information systems. In *Proc. of ECAI*. IOS Press, 2008.

[9] B. Cuenca Grau, B. Motik, and Y. Kazakov. Import-by-Query: Ontology Reasoning under Access Limitations. In *Proc. IJCAI*, pages 727–733. AAAI Press, 2009.

[10] A. Deutsch and Y. Papakonstantinou. Privacy in database publishing. In *Proc. of ICDT*, pages 230–245, 2005.

[11] A. V. Evfimievski, R. Fagin, and D. P. Woodruff. Epistemic privacy. In *Proc. of PODS-08*, pages 171–180. ACM, 2008.

[12] W. Fan, C. Y. Chan, and M. N. Garofalakis. Secure xml querying with security views. In *Proc. of SIGMOD*, pages 587–598, 2004.

[13] B. C. Grau and B. Motik. Pushing the Limits of Reasoning over Ontologies with Hidden Content. In *Proc. of the 12th Int. Conference on Principles of Knowledge Representation and Reasoning (KR 2010)*, Toronto, ON, Canada, May 9–13 2010. AAAI Press. To appear.

[14] A. Y. Halevy. Answering queries using views: A survey. *VLDB J.*, 10(4):270–294, 2001.

[15] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a Web ontology language. *J. of Web Semantics, Elsevier*, 1(1):7–26, 2003.

[16] V. Kolovski, J. A. Hendler, and B. Parsia. Analyzing Web access control policies. In *Proc. of WWW*, pages 677–686, 2007.

[17] B. Konev, D. Walter, and F. Wolter. Forgetting and uniform interpolation in large-scale description logic terminologies. In *Proc. IJCAI*. AAAI Press, 2009.

[18] A. Y. Levy. Obtaining complete answers from incomplete databases. In *Proc. of VLDB*, pages 402–412, 1996.

[19] A. Machanavajjhala and J. Gehrke. On the efficiency of checking perfect privacy. In *Proc. of PODS 2006*, 2006.

[20] G. Miklau and D. Suciu. A formal analysis of information disclosure in data exchange. *J. Comput. Syst. Sci.*, 73(3):507–534, 2007.

[21] T. Moses. Oasis Extensible Access Control Markup Language. Oasis Standard, 2005.

[22] B. Motik, I. Horrocks, and U. Sattler. Bridging the Gap Between OWL and Relational Databases. In *Proc. of WWW 2007*, pages 807–816. ACM Press, 2007.

[23] S. Rizvi, A. O. Mendelzon, S. Sudarshan, and P. Roy. Extending query rewriting techniques for fine-grained access control. In *Proc. of SIGMOD-04*, pages 551–562. ACM, 2004.

[24] P. Stouppa and T. Studer. A formal model of data privacy. In *Pro. of PSI-06*, volume 4378 of *LNCS*. Springer, 2007.

[25] P. Stouppa and T. Studer. Data privacy for knowledge bases. In *LFCS*, pages 409–421, 2009.

[26] A. Vedamuthu. Web Services Policy 1.5 - Framework. World Wide Web Consortium (W3C) Recommendation, 2007.

[27] N. Zhang, M. Ryan, and D. Guelev. Evaluating access control policies through model checking. In *Proc. of ISC*, 2005.