A Formal Model of Data Privacy

Phiniki Stouppa and Thomas Studer

Institut für Informatik und angewandte Mathematik,
Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland,
{stouppa,tstuder}@iam.unibe.ch

Abstract. Information systems support data privacy by constraining user's access to public views and thereby hiding the non-public underlying data. The privacy problem is to prove that none of the private data can be inferred from the information which is made public. We present a formal definition of the privacy problem which is based on the notion of certain answer. Then we investigate the privacy problem in the contexts of relational databases and ontology based information systems.

1 Introduction

The development of automatic information processing has made it necessary to consider privacy protection in relation to personal data. The surveillance potential of powerful computer systems demands for rules governing the collection and sharing of personal information. An overview of the evolution of data protection is presented in [18].

Two of the main international instruments in this context are the Council of Europe's 1981 Convention for Protection of Individuals with regard to Automatic Processing of Personal Data [7] and the Organisation for Economic Cooperation and Development (OECD) Guidelines on the Protection of Privacy and Transborder Flows of Personal Data [17]. These rules describe *personal data* as any information relating to an identified or identifiable individual.

The expression of data protection in various declarations and laws varies. However, all require that personal data must be kept secure. That includes appropriate security measures for the protection of personal data stored in information systems against unauthorized access. Thus, information systems must take responsibility for the data they manage [1]. The main challenge in data privacy is to share some data while protecting personal information.

We provide a theoretical framework to prove that under certain circumstances none of the personal data can be inferred from the information which is made public. The underlying system is given in the form of an

ontology. Personal data takes the form of a *privacy condition* which is a set of queries. Moreover, the public information is given in terms of a *view instance* and *background knowledge*. A view instance consists of queries and their (actual) answers, while background knowledge includes additional facts about the system that are provided for better understanding of the data in the views. The *privacy problem* is then to decide whether any of the queries in the private condition can be inferred from the view instance and the background knowledge.

In order to state the privacy problem, we employ the notion of *certain answer*: data privacy is preserved for a query with respect to the provided public knowledge if there are no non-negative certain answers of the query with respect to that knowledge. That is, if the certain answer to it is either the empty set or negative ("None" or "No"). The certain answers of a query are those answers that are returned by the query in every 'possible' instance. The problem of answering queries against a set of 'possible' instances was first encountered in the context of incomplete databases [22]. Today, certain answer is a key notion in the theory of data integration [6, 13, 15] and data exchange [2, 12].

Let us demonstrate the above setting: consider an ontology that contains information about the customers of a telecommunication company. The company provides information to the end-users through searching engines on its telephone lists, whereas at the same time some of its customers do not wish to give in publicity their telephone numbers. Thus, the privacy condition would be a set of queries of the form $\mathsf{Owns}(\mathsf{cust}_i, \mathsf{Tel})$, where Owns relates customers to their telephone numbers, cust_i is a constant and Tel is a variable. Since these are retrieval queries, data privacy is preserved when there is no certain answer to each of them. That is, there is no telephone number which is returned by such a query in every 'possible' ontology. If this holds, then the set of certain answers is empty which means that no telephone number of any of cust_is is exhibited. Negative answers might occur only in the case of boolean queries that are not applicable on the ontology, when this is also announced through the public information.

Our work is concerned with the question how much information a given view instance reveals and whether it leaks private data. Much of the existing work on privacy for information systems deals with privacy preserving query answering. There, the privacy problem is that of inferring a maximal subset of the answer to a query so that no secrets are violated [5, 23]. The idea of specifying sensitive information as conjunctive query is pioneered in [16], where the notion of perfect privacy is

introduced. However, enforcing perfect privacy for conjunctive queries is very intractable. A generalization of this model has been studied in [10]. There, checking perfect privacy is even harder. Recently, Dix et al. [11] established a relationship between privacy problems and non-monotonic logics. Another approach [20] is to generalize the answers to a query in order to provide anonymity.

The rest of the paper is organized as follows: first, we give formal definitions for both the ontology and query answering on it. We define the ontology as a set of first-order sentences, while query answering is done via entailment. This allows for the application of data privacy in both knowledge base and database systems. Thus, the present definition of data privacy is much more general than the one given in [21] which applies to relational databases only. Then, we present a formal model of data privacy using certain answers and show that these can be reduced to logical entailment. Thus, in general, the privacy problem is not decidable. We continue by presenting two applications where the data privacy problem is decidable: in Section 4 we apply data privacy on relational databases with conjunctive queries. In this case, background knowledge consists of a relational schema with constraints imposed on it. Data privacy for this setting is decidable in polynomial time. In Section 5 we apply data privacy on $\mathcal{ALC}$ description logic-based ontologies. In this case, background knowledge might include any TBox or ABox entries. Here, the complexity of data privacy follows the complexity of $\mathcal{ALC}$-reasoning: it is ExpTime-complete for ontologies with a general TBox and PSpace-complete for ontologies with an acyclic TBox. Finally, we summarize the results and give further research directions.

2 The Ontology and Query Answering

We define the *relational first-order language* $\mathcal{L}$ as follows. The collection of $\mathcal{L}$ terms comprises countably many *variables* $x, y, z, \ldots$ and countably many *constant symbols* $a, b, c, \ldots$. We use Const for the set of $\mathcal{L}$ constants. $\mathcal{L}$ includes for every natural number n countably many *relation symbols* $R, S, T, \ldots$ of arity n as well as the binary relation symbol $=$ for equality. If R is an n-ary relation symbol of $\mathcal{L}$ and $t_1, \ldots, t_n$ are $\mathcal{L}$ terms, then $R(t_1, \ldots, t_n)$ is an *atomic* $\mathcal{L}$ *formula*. $\mathcal{L}$ *formulae* are built up inductively from the atomic formulae of $\mathcal{L}$ by closing under the usual connectives as well as universal and existential quantification. We call an $\mathcal{L}$ formula without free variables $\mathcal{L}$ *sentence*.

We will also make use of the standard notion of *logical entailment*: Let ϕ be a formula and $\mathcal{O}$ a set of formulae. Then $\mathcal{O} \models \phi$ if every model of $\mathcal{O}$ is also a model of ϕ.

Note that the choice of a first-order language for the current presentation is not important. We could as well use any other language that is employed in the context of information systems, such as second order languages or fixed point logics. Now, we can formally introduce the ontology and show how query answering can be defined in terms of entailment:

Definition 1. *An* ontology $\mathcal{O}$ *is a finite set of* $\mathcal{L}$ *sentences.* $\mathsf{Const}(\mathcal{O})$ *denotes the set of constants that occur in* $\mathcal{O}$*. A* query q *is an* $\mathcal{L}$ *formula. If* q *has no free variables, then* q *is called* boolean query *otherwise it is a* retrieval query.

Definition 2. *The* range of a query q ($\mathsf{range}(q)$) *is given by:*

1. $\{\emptyset, \{\top\}, \{\bot\}\}$ *if* q *is a sentence,*
2. $\mathsf{Pow}(\mathsf{Const}^n)$ *which is the power set of the* n *times Cartesian product of* Const *with itself, if* q *is a formula with* $n > 0$ *free variables.*

Definition 3. *The* answer to a query q with respect to an ontology $\mathcal{O}$ ($\mathsf{ans}(q, \mathcal{O})$) *is given by:*

$\mathsf{ans}(q, \mathcal{O}) := \{\top\}$ *if* q *is a sentence and* $\mathcal{O} \models q$*,*
$\mathsf{ans}(q, \mathcal{O}) := \{\bot\}$ *if* q *is a sentence,* $\mathcal{O} \not\models q$ *and* $\mathcal{O} \models \neg q$*,*
$\mathsf{ans}(q, \mathcal{O}) := \emptyset$ *if* q *is a sentence,* $\mathcal{O} \not\models q$ *and* $\mathcal{O} \not\models \neg q$*,*
$\mathsf{ans}(q, \mathcal{O}) := \{\boldsymbol{t} \in \mathsf{Const}(\mathcal{O})^n \mid \mathcal{O} \models q(\boldsymbol{t})\}$ *if* q *has* $n > 0$ *free variables.*

Note that $\mathsf{ans}(q, \mathcal{O}) \in \mathsf{range}(q)$ and is always finite. Finally, a view instance is a set of queries together with their answers:

Definition 4. *A* view instance V_I *is a finite set of tuples* $\langle q_i, r_i \rangle$ *where each* q_i *is a query and* $r_i \in \mathsf{range}(q_i)$*. We say that an* ontology $\mathcal{O}$ entails a view instance V_I *(in symbols* $\mathcal{O} \models V_I$*) if* $r_i = \mathsf{ans}(q_i, \mathcal{O})$ *for every* $\langle q_i, r_i \rangle \in V_I$*.*

3 Data Privacy

As mentioned in the introduction, in addition to the view instance V_I that is provided, public knowledge also includes some other facts, the background knowledge. We will refer to it as the ontology $\mathcal{O}$. We call

the tuple $\langle \mathcal{O}, V_I \rangle$ a *data privacy setting*. Also, since querying an ontology makes sense only when the answers it provides do actually hold, we assume that the underlying ontology is consistent.

We give a definition of the problem based on the notion of certain answer: let q be the information we wish to keep private. First, we collect all those ontologies each of which is conceivably the underlying ontology. Afterwards, we collect those answers to q that do *certainly* hold in each of the collected ontologies. A non-negative answer would then mean that q is exhibited and thus, data privacy is not preserved.

Definition 5. *Let $\langle \mathcal{O}, V_I \rangle$ be a data privacy setting. We call an ontology $\mathcal{P}$* possible *with respect to $\langle \mathcal{O}, V_I \rangle$ if*

1. *$\mathcal{P}$ is consistent,*
2. *$\mathcal{O} \subseteq \mathcal{P}$, and*
3. *$\mathcal{P} \models V_I$.*

$\mathsf{Poss}_{\langle \mathcal{O}, V_I \rangle}$ denotes the set of all possible ontologies with respect to $\langle \mathcal{O}, V_I \rangle$.

Definition 6. *The* certain answers *to a query q with respect to a setting $\langle \mathcal{O}, V_I \rangle$ are defined by*

$$\mathsf{certain}(q, \langle \mathcal{O}, V_I \rangle) := \bigcap_{\mathcal{P} \in \mathsf{Poss}_{\langle \mathcal{O}, V_I \rangle}} \mathsf{ans}(q, \mathcal{P})$$

Definition 7. *We say* data privacy is preserved for q with respect to $\langle \mathcal{O}, V_I \rangle$ *if* $\mathsf{certain}(q, \langle \mathcal{O}, V_I \rangle) \subseteq \{\bot\}$.

The proposed definition has the advantage that works independently of the underlying language. However, it does not provide a direct solution to the problem as the possible ontologies are infinitely many. For this reason, we first construct a so-called *canonical* ontology that carries minimal, though complete, information about the certain answers to a given query.

Definition 8. *Given a setting $\langle \mathcal{O}, V_I \rangle$, the canonical ontology $\mathcal{C}_{\langle \mathcal{O}, V_I \rangle}$ is defined as*

$$\begin{aligned} \mathcal{C}_{\langle \mathcal{O}, V_I \rangle} := \mathcal{O} \cup \\ & \{q \mid \langle q, \{\top\} \rangle \in V_I\} \cup \\ & \{\neg q \mid \langle q, \{\bot\} \rangle \in V_I\} \cup \\ & \{q(\boldsymbol{t}) \mid \textit{there is an } A \textit{ with } \langle q, A \rangle \in V_I \textit{ and } \boldsymbol{t} \in A\} \end{aligned}$$

Note that this construction is language-dependent. The following theorem can be easily shown:

Theorem 1. *Given an $\mathcal{L}$ formula ϕ and a data privacy setting $\langle \mathcal{O}, V_I \rangle$, the following holds:*

$$\mathcal{C}_{\langle \mathcal{O}, V_I \rangle} \models \phi \text{ if and only if } \forall \mathcal{P}.(\mathcal{P} \in \mathsf{Poss}_{\langle \mathcal{O}, V_I \rangle} \rightarrow \mathcal{P} \models \phi).$$

In order to check whether data privacy is preserved for a query q with respect to $\langle \mathcal{O}, V_I \rangle$, we can build the canonical ontology $\mathcal{C}_{\langle \mathcal{O}, V_I \rangle}$ and issue q to it.

Corollary 1. *Data privacy is preserved for q with respect to $\langle \mathcal{O}, V_I \rangle$ if and only if $\mathsf{ans}(q, \mathcal{C}_{\langle \mathcal{O}, V_I \rangle}) \subseteq \{\bot\}$.*

4 Relational Databases

In this section we show that there is a polynomial time solution to the privacy problem for relational databases. Although classical database theory is concerned with model checking, we can make use of Reiter's proof theoretic approach [19] in order to apply our setting to relational databases.

In the context of relational databases, we consider only conjunctive queries.

Definition 9. *An $\mathcal{L}$ formula is called* conjunctive query *if it is built from atomic formulae, conjunctions and existential quantifiers. A* conjunctive view instance *V_I is a view instance such that q_i is a conjunctive query for each $\langle q_i, r_i \rangle \in V_I$.*

Definition 10. *A* data privacy setting for databases *$\langle \mathcal{O}, V_I \rangle$ consists of*

1. *a set of dependencies $\mathcal{O}$. Each element of $\mathcal{O}$ is either a tuple generating dependency [4] of the form*

   $$\forall \boldsymbol{x}(\phi(\boldsymbol{x}) \rightarrow \exists \boldsymbol{y} \psi(\boldsymbol{x}, \boldsymbol{y}))$$

   or an equality generating dependency [4] of the form

   $$\forall \boldsymbol{x}(\phi(\boldsymbol{x}) \rightarrow (x_1 = x_2)),$$

   where $\phi(\boldsymbol{x})$ and $\psi(\boldsymbol{x}, \boldsymbol{y})$ are conjunctions of atomic formulae and x_1, x_2 are among the variables of $\boldsymbol{x}$,
2. *a conjunctive view instance V_I.*

It is possible to translate the data privacy setting for databases to a data exchange setting [21]. Fagin et al. [12] show that in such a setting, the classical chase can be used to compute certain answers for conjunctive queries. The procedure they present terminates in polynomial time.

Theorem 2. *Given a data privacy setting for databases $\langle \mathcal{O}, V_I \rangle$ and a conjunctive query q. Then we can check in polynomial time whether privacy is preserved for q with respect to $\langle \mathcal{O}, V_I \rangle$.*

5 $\mathcal{ALC}$-based Ontologies

Description logics build the mathematical core of many modern knowledge base systems [3]. Their language consists of *concepts* (sets of individuals) and *roles* (binary relationships between the individuals).

The basic description logic $\mathcal{ALC}$ consists of the following concepts:

$$C := A \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \forall R.C \mid \exists R.C,$$

where A is an atomic concept and R is a role. Each concept C abbreviates an $\mathcal{L}$ formula $C'(x)$ with one free variable x as follows.

$$\begin{aligned} A'(x) &:= A(x) \\ (\neg C)'(x) &:= \neg C'(x) \\ (C_1 \sqcap C_2)'(x) &:= C_1'(x) \wedge C_2'(x) \\ (C_1 \sqcup C_2)'(x) &:= C_1'(x) \vee C_2'(x) \\ (\forall R.C)'(x) &:= \forall y.(R(x,y) \rightarrow C'(y)) \\ (\exists R.C)'(x) &:= \exists y.(R(x,y) \wedge C'(y)) \end{aligned}$$

In the sequel, we will identify concepts and the corresponding $\mathcal{L}$ formulae. An ontology contains a terminology, that is the vocabulary of an application domain, as well as assertions about named individuals in terms of the vocabulary. The *terminology* consists of concept equality axioms of the form $C_1 \equiv C_2$ abbreviating $\forall x.(C_1(x) \leftrightarrow C_2(x))$. An *assertion* is a formula of the form $C(a)$ or $R(a,b)$ where $a, b \in \mathsf{Const}$ are called *individuals*. An *$\mathcal{ALC}$-based ontology* consists of a terminology (called TBox) and a set of assertions (called ABox).

A TBox is *acyclic* when it satisfies the following: (i) every concept equality is of the form $A \equiv C$, (ii) every atomic formula occurs at most once at the left hand side of an equality and (iii) there are no cycles in the concept equality axioms.

An $\mathcal{ALC}$ *query* is either a concept (retrieval query) or an expression of the form $C(a)$ or $C_1 \equiv C_2$ (boolean query).[1] A setting $\langle \mathcal{O}, V_I \rangle$ is a data

[1] The problems of querying a concept assertion and querying an equality are known as the instance and equivalence problems, respectively. The well-known subsumption problem is reduced to the equivalence problem.

privacy setting for $\mathcal{ALC}$-based ontologies if $\mathcal{O}$ is an $\mathcal{ALC}$-based ontology and V_I is given by $\mathcal{ALC}$ queries. For the rest of this section, *query* refers to $\mathcal{ALC}$ query.

The data privacy problem in this setting can be solved following the approach presented in the general setting, that is, by building a canonical ontology that corresponds to the public knowledge $\langle \mathcal{O}, V_I \rangle$. In its current form, the ontology defined in Definition 8 is not an $\mathcal{ALC}$-based ontology, since a negative answer on an equality query $C_1 \equiv C_2$ would include a non-$\mathcal{ALC}$ formula. What actually a negative answer tells about the ontology in this case, is that there is an individual which belongs to C_1 and does not belong to C_2 or vice versa. Thus, we can unfold the view instance by replacing every $\langle C_1 \equiv C_2, \{\bot\}\rangle$ in V_I by $\langle (C_1 \sqcap \neg C_2) \sqcup (\neg C_1 \sqcap C_2)(d), \{\top\}\rangle$, where d is fresh (that is it does not occur in $\langle \mathcal{O}, V_I \rangle$ or in the private query q). We can now construct the canonical ontology based on this unfolded view instance.

Similarly to Theorem 1, it can be shown that the constructed ontology is indeed canonical with respect to the public knowledge. Finally, under this framework, the complexity results for the reasoning problem in $\mathcal{ALC}$-based ontologies [3] apply also to the privacy problem.

Theorem 3. *Given a data privacy setting $\langle \mathcal{O}, V_I \rangle$ for $\mathcal{ALC}$-based ontologies and a query q, the data privacy problem for q with respect to $\langle \mathcal{O}, V_I \rangle$ is* ExpTime*-complete when the* TBox *in $\langle \mathcal{O}, V_I \rangle$ is general and* PSpace*-complete when it is acyclic.*

Note that in the context of description logic ontologies, our approach is not restricted to $\mathcal{ALC}$. We can use the same method also to solve the data privacy problem for ontologies which are given in very expressive description logics. For instance, our technique also applies to logics such as $\mathcal{SHIF}$ and $\mathcal{SHOIN}$ which are the mathematical models for the web ontology languages OWL Lite and OWL DL.

However, if the query language is different from the ontology language, then Definition 8 is not applicable. For instance, if we have a description logic based ontology language and use conjunctive queries to retrieve information, then we need other techniques to solve the privacy problem.

6 Conclusion and Outlook

We have given a formal definition of the general data privacy problem for information systems. This problem is to check whether a given view instance leaks information about the underlying data or knowledge base.

We have modeled the privacy problem using the notion of certain answer. Privacy holds for a query q with respect to a view instance V_I if there are no non-negative certain answers to q with respect to V_I.

Computing certain answers is equivalent to logical entailment. Thus it is in general undecidable. We have investigated two important decidable cases: the privacy problem for relational databases with a set of constraints and the privacy problem for ontology (description logic) based information systems.

We plan to extend our study to other data models. The investigation of the privacy problem for XML databases is an important further task. Like relational databases, XML databases protect data from unauthorized access by allowing users to issue queries solely to views that provide public information only [9]. The computation of certain answers in XML databases has been studied for instance in [2].

Another direction of future work is to investigate the effect of updates to data privacy. Assume we have a query and a view instance for which privacy holds. If we update the underlying database or ontology, can we be sure that privacy still is preserved? Thus, it is important to study privacy preserving updates. That is, the question of which forms of updates do not violate data privacy.

The present definition of the privacy problem consists of deciding whether a given view instance leaks information. There is a second privacy problem: deciding whether already the view definition guarantees that there is no possible leaking. That means, given the view definition, there cannot be a view instance that leaks private information. For example, this is the case in relational databases if values stored in private attributes cannot be inferred via the constraints defined in the database. In ontology based systems, the theory of $\mathcal{E}$-connections [14] and partitioning of ontologies [8] may lead to such secure view definitions. Finally, the study of this second privacy problem will result in a collection of database patterns which are safe with respect to data privacy.

Acknowledgments

We would like to thank Sebastian Link for bringing our attention to the privacy problem for XML databases.

References

1. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *Proc. of 28th VLDB Conference*, 2002.

2. M. Arenas and L. Libkin. XML data exchange: Consistency and query answering. In *PODS*, pages 13–24, 2005.
3. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
4. C. Beeri and M. Y. Vardi. A proof procedure for data dependencies. *Journal of the ACM*, 31(4):718–741, 1984.
5. P. A. Bonatti, S. Kraus, and V. s. Subrahmanian. Foundations of secure deductive databases. *Transactions on Knowledge and Data Engineering*, 7(3):406–422, 1995.
6. A. Calì, D. Calvanese, G. D. Giacomo, and M. Lenzerini. Data integration under integrity constraints. In *Proc. of CAiSE 2002*, volume 2348 of *LNCS*, pages 262–279. Springer, 2002.
7. Council of Europe. Convention for the protection of individuals with regard to automatic processing of personal data, 1981. Available at `http://conventions.coe.int/Treaty/en/Treaties/Html/108.htm`.
8. B. Cuenca Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Automated partitioning of owl ontologies using e-connections. In *Proceedings of Int. Workshop on Description Logics*, 2005.
9. E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Controlling access to XML documents. *IEEE Internet Computing*, 5(6):18–28, 2001.
10. A. Deutsch and Y. Papakonstantinou. Privacy in database publishing. In *ICDT*, 2005.
11. J. Dix, W. Faber, and V. Subrahmanian. The relationship between reasoning about privacy and default logics. In *LPAR*, pages 637–650. Springer, 2005.
12. R. Fagin, P. G. Kolaitis, R. Miller, and L. Popa. Data exchange: Semantics and query answering. *Theoretical Computer Science*, 336:89–124, 2005.
13. A. Y. Halevy. Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294, 2001.
14. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of abstract description systems. *Artifical Intelligence*, 156(1):1–73, 2004.
15. M. Lenzerini. Data integration: a theoretical perspective. In *ACM PODS '02*, pages 233–246. ACM Press, 2002.
16. G. Miklau and D. Suciu. A formal analysis of information disclosure in data exchange. In *SIGMOD*, 2004.
17. OECD. Guidelines on the protection of privacy and transborder flows of personal data, 1980. Available at `http://www.oecd.org`.
18. Privacy International. Overview of privacy, 2004. Available at `http://www.privacyinternational.org/privhroverview2004`.
19. R. Reiter. Towards a logical reconstruction of relational database theory. In M. Brodie, J. Mylopoulos, and J. Schmidt, editors, *On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages*, pages 191–233. 1982.
20. P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *PODS*, page 188. ACM Press, 1998.
21. K. Stoffel and T. Studer. Provable data privacy. In K. Viborg, J. Debenham, and R. Wagner, editors, *Database and Expert Systems Applications DEXA 2005*, volume 3588 of *LNCS*, pages 324–332. Springer, 2005.
22. R. van der Meyden. Logical approaches to incomplete information: a survey. In *Logics for databases and information systems*, pages 307–356. Kluwer Academic Publishers, 1998.
23. M. Winslett, K. Smith, and X. Qian. Formal query languages for secure relational databases. *ACM Trans. Database Syst.*, 19(4):626–662, 1994.