Privacy Preservation Using Multi-Context Systems*

Wolfgang Faber

University of Calabria, Italy
wf@wfaber.com

Abstract. Preserving the privacy of sensitive data is one of the major challenges which the information society has to face. Traditional approaches focused on the infrastructure for identifying data which is to be kept private and for managing access rights to these data. However, while these efforts are useful, they do not address an important aspect: While the sensitive data itself can be protected nicely using these mechanisms, related data, which is deemed insensitive per se may be used to infer sensitive data. This can be achieved by combining insensitive data or by exploiting specific background knowledge of the domain of discourse. In this note, we show that resolving this problem can be achieved in a simple and elegant way by using multi-context systems.

1 Introduction

The privacy of individuals has become one of the most important and most discussed issues in modern society. With the advent of the Internet and easy access to a lot of data, keeping sensitive data private has become a priority for distributed information systems. An example area in which privacy is at stake are medical information systems.

Most databases have privacy mechanisms which are comparatively simple – by and large they boil down to keeping certain columns of the database hidden from certain types of users. There is a huge body of literature that deals with formalisms for this kind of authorization problem, which we cannot discuss in detail in this short note. As an example, see [6] for a work that discusses aspects of the authorization problem in non-monotonic knowledge bases. What we are interested in this short paper is a somewhat different issue, namely that users can infer information that is designated private by asking queries that do not involve private information and then making "common sense" inferences from the answers to infer private information.

In an earlier paper [4], we have given a formal definition of the *Privacy Preservation Problem* and shown how this can be addressed by using default logic (we also refer to this paper for discussions on related work). In that paper, however, there were several restrictions on the knowledge bases that can be used. Effectively, they had to be first-order theories, because in this way it is easily possible to build a default theory around them.

In order to lift this restriction, in this note we propose using multi-context systems as defined by Brewka and Eiter in [3] instead of default logic. By switching to that formalism, it is possible to use heterogeneous knowledge bases to which users may

* This work was supported by M.I.U.R. within the PRIN project LoDeN.

have access or which model user knowledge. The unifying framework are then contexts and bridge rules that link contexts instead of default rules in [4]. Apart from the greater flexibility concerning the types of "participating" knowledge bases, another advantage is that efficient systems for reasoning with multi-context systems begin to emerge [1].

In the following, we will first provide an adapted definition of the privacy preservation problem in section 2. This definition is slightly different from the one of [4] in order to allow for more heterogeneous knowledge bases to be involved. In section 3 we will then show how to construct a multi-context system for computing answers for a privacy preservation problem. In section 4 we conclude and outline future work.

2 Privacy Preservation Problem

In this section, we provide a simple formulation of the privacy preservation problem (**P3** for short), which is a generalization of the definition in [4]. For simplicity, we will not consider any evolution in time of the systems, as it was done in [4].

We consider a *logic* L as in [3] to be a triple $(\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{ACC}_L)$ where $\mathbf{KB}_L$ is the set of well-formed knowledge bases of L (each of which is a set as well), $\mathbf{BS}_L$ is the set of possible belief sets, and $\mathbf{ACC}_L$ is a function $\mathbf{KB}_L \rightarrow \mathbf{2}^{\mathbf{BS}_L}$ describes the semantics of each knowledge base. In the following, when mentioning knowledge bases, we will usually not specify the underlying logic, intending that it can be any logic in the sense just described.

Let the finite set $\mathbf{U}$ contain one user ID for each user in the system under consideration. Moreover, we consider the *main knowledge base* $\mathbf{MKB}$ which the users will be querying. Furthermore, the function $\mathbf{BK}$ associates each user $u \in \mathbf{U}$ with a *background knowledge base* $\mathbf{BK}(u)$, while the function $\mathbf{Priv}$ associates each user $u \in \mathbf{U}$ with a belief set $\mathbf{Priv}(u)$ that should be kept private. Note that the various knowledge bases need not be of the same logic, but for practical reasons one would assume the belief sets to be homogeneous.

It should be pointed out that $\mathbf{BK}(u)$ is not necessarily the user's own knowledge base, but rather a model of the user's knowledge, maintained by the information system.

Example 1. Consider a small medical knowledge base MedKB containing information about the symptoms and diseases of some patients. Let this knowledge base describe two predicates symptom and disease and let the following be its only belief set S_{MedKB}:

$\mathsf{symptom}(john, s_1)$	$\mathsf{symptom}(jane, s_1)$	$\mathsf{disease}(jane, aids)$
$\mathsf{symptom}(john, s_2)$	$\mathsf{symptom}(jane, s_4)$	$\mathsf{disease}(john, cancer)$
$\mathsf{symptom}(john, s_3)$		$\mathsf{disease}(ed, polio)$

Note that MedKB could very well be just a database. Assume that $john$ and $jane$ are also users of the system and want to keep their diseases private, so $\mathbf{Priv}(john) = \{\mathsf{disease}(john, cancer)\}$, while $\mathbf{Priv}(jane) = \{\mathsf{disease}(jane, aids)\}$. Consider another user $acct$ (an accountant). This person may have the following background knowledge base $\mathbf{BK}(acct)$ in the form of rules (so the underlying logic might be answer set programming).

$\mathsf{disease}(X, aids) \leftarrow \mathsf{symptom}(X, s_1), \mathsf{symptom}(X, s_4)$
$\mathsf{disease}(X, cancer) \leftarrow \mathsf{symptom}(X, s_2), \mathsf{symptom}(X, s_3)$

Let a query be a construct to which for every semantics of a knowledge base a belief set is associated, which is referred to as the *answer* $\mathbf{Ans}(Q)$ to Q. A *privacy preserving answer* to a query Q over $\mathbf{MKB}$ posed by $u_o \in \mathbf{U}$ with respect to $\mathbf{BK}$ and $\mathbf{Priv}$ is $X \subseteq \mathbf{Ans}(Q)$ such that for all $u \in \mathbf{U} \setminus \{u_0\}$ and for all $p \in \mathbf{Priv}(u)$, if $p \notin \mathbf{ACC}(\mathbf{BK}(u_0))$ then $p \notin \mathbf{ACC}(X \cup \mathbf{BK}(u_0))$. A *maximal privacy preserving answer* is a subset maximal privacy preserving answer.

Note that here we assume that elements of belief sets can be added to knowledge bases, yielding again a knowledge base of the respective logic.

A privacy preservation problem $\mathbf{P3}$ is therefore a tuple $(\mathbf{MKB}, \mathbf{U}, \mathbf{BK}, \mathbf{Priv}, Q, u_0)$ and solving it amounts to finding the (maximal) privacy preserving answers to Q posed by u_0 over $\mathbf{MKB}$ with respect to $\mathbf{BK}$ and $\mathbf{Priv}$.

Example 2. Returning to our MedKB example, posing the query $\mathsf{disease}(john, X)$, we would get as an answer the set $\{\mathsf{disease}(john, cancer)\}$. Likewise, the answer to the query $\mathsf{symptom}(john, X)$ is the set $\{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_2), \mathsf{symptom}(john, s_3)\}$.

We assumed that John and Jane want their diseases kept private. However, the accountant can violate John's privacy by asking the query $\mathsf{symptom}(john, X)$. The answer that $acct$ would get from the system is $\{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_2), \mathsf{symptom}(john, s_3)\}$. However, recall that the accountant has some background knowledge including the rule

$$\mathsf{disease}(X, cancer) \leftarrow \mathsf{symptom}(X, s_2), \mathsf{symptom}(X, s_3)$$

which, with the answer of the query, would allow $acct$ to infer $\mathsf{disease}(john, cancer)$. Thus the privacy preserving answers to $\mathsf{symptom}(john, X)$ are

$$\begin{aligned}
Ans_1 &= \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_2)\} \\
Ans_2 &= \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_3)\} \\
Ans_3 &= \{\mathsf{symptom}(john, s_1)\} \\
Ans_4 &= \{\mathsf{symptom}(john, s_2)\} \\
Ans_5 &= \{\mathsf{symptom}(john, s_3)\} \\
Ans_6 &= \emptyset
\end{aligned}$$

None of these answers allows $acct$ to infer the private knowledge $\mathsf{disease}(john, cancer)$. However, except for the answers Ans_1 and Ans_2, which are maximal, all answers yield fewer information than could be disclosed without infringing privacy requirements. Any system should also provide only one of these answers to the user, because getting for instance both Ans_1 and Ans_2 would again violate John's privacy requirements.

In a practical system, upon disclosing an answer the system should update the respective user's knowledge model in order to avoid privacy infringements by repeated querying. For example, when the system returns Ans_1 to user $acct$, it should modify $\mathbf{BK}(acct)$ in order to reflect the fact that $acct$ now knows $\mathsf{symptom}(john, s_1)$ and $\mathsf{symptom}(john, s_2)$, such that asking the same query again it is made sure that $\mathsf{symptom}(john, s_3)$ will not be disclosed to $acct$. This however is part of the dynamic aspect of a privacy preserving information system, which we will not address in this paper.

3 Solving Privacy Preservation Problems Using Multi-Context Systems

The definitions in Section 2 were already slightly geared towards multi-context systems. We recall that a multi-context system in the sense of [3] is a tuple $(C_1, \ldots, C_n)$ where for each i, $C_i = (L_i, kb_i, br_i)$ where L_i is a logic, kb_i is a knowledge base of L_i and br_i is a set of L_i bridge rules over $\{L_1, \ldots, L_n\}$.

An L_i bridge rule over $\{L_1, \ldots, L_n\}$ is a construct

$$s \leftarrow (r_1 : p_1), \ldots, (r_j : p_j), \text{not } (r_{j+1} : p_{j+1}), \ldots, \text{not } (r_m : p_m)$$

where $1 \leq r_k \leq n$, p_k is an element of a belief set for L_{r_k} and for each $kb \in \mathbf{KB}_i$ $kb \cup \{s\} \in \mathbf{KB}_i$.

The semantics of a multi-context system is defined by means of equilibria. A *belief state* for a multi-context system $(C_1, \ldots, C_n)$ is $S = (S_1, \ldots, S_n)$, where $S_i \in \mathbf{BS}_i$ for $1 \leq i \leq n$. An L_i bridge rule of the form above is applicable in S iff for $1 \leq k \leq j$ $p_k \in S_{r_k}$ holds and for $j < k \leq m$ $p_k \notin S_{r_k}$ holds. Let $app(br, S)$ denote the set of all bridge rules in br which are applicable in a belief state S. A belief state $S = (S_1, \ldots, S_n)$ is an equilibrium of a multi-context system $(C_1, \ldots, C_n)$ iff for all $1 \leq i \leq n$, $S_i \in \mathbf{ACC}_i(kb_i \cup \{hd(r) \mid r \in app(br_i, S)\})$, where $hd(r)$ is the head of a bridge rule r, viz. s in the bridge rule schema given earlier.

Given a $\mathbf{P3}$ $(\mathbf{MKB}, \mathbf{U}, \mathbf{BK}, \mathbf{Priv}, Q, u)$, with $\mathbf{U} = \{u_1, \ldots, u_{|\mathbf{U}|}\}$, in order to identify privacy preserving answers, we build a multi-context system $M_{\mathbf{P3}} = (C_1, C_2, C_3, C_4, \ldots, C_{|\mathbf{U}|+3})$, where $C_1 = (L_{\mathbf{MKB}}, \mathbf{MKB}, \emptyset)$, $C_2 = (L_{\mathbf{MKB}}, \emptyset, br_2)$, $C_3 = (L_{\mathbf{MKB}}, \emptyset, br_3)$, $C_4 = (L_{\mathbf{BK}(u_1)}, \mathbf{BK}(u_1), br_4) \ldots$, $C_{|\mathbf{U}|+3} = (L_{\mathbf{BK}(u_{|\mathbf{U}|})}, \mathbf{BK}(u_{|\mathbf{U}|}), br_{|\mathbf{U}|+3})$. Here L_{kb} is the logic of the knowledge base kb. The meaning is that C_1 provides just the belief sets for $\mathbf{MKB}$ (no bridge rules), C_2 and C_3 are used to identify those belief sets which are privacy preserving, while $C_4, \ldots, C_{|\mathbf{U}|+3}$ represent the user information, that is, the background knowledge base of the querying user and the privacy requirements of the other users. The important part are the bridge rules, which we will describe next. In many cases, we will create one rule for each symbol that can occur in some belief set of $\mathbf{Ans}(Q)$, so for convenience let $\mathcal{D} = \{p \mid p \in B, B \in \mathbf{Ans}(Q)\}$.

The set br_2 contains one bridge rule $p \leftarrow (1 : p), \text{not } (3 : p)$ for each $p \in \mathcal{D}$. Symmetrically, br_3 contains one bridge rule $p \leftarrow (1 : p), \text{not } (2 : p)$ for each $p \in \mathcal{D}$. The intuition is that the belief sets of C_2 will be subsets of the belief set of C_1 in any equilibrium, and hence possible privacy preserving answers. C_3 exists only for technical reasons.

For i such that $u_{i-2} = u$, thus for the context C_i of the querying user, we add one bridge rule $p \leftarrow (2 : p)$ for each $p \in \mathcal{D}$. This means that in any equilibrium the belief set for i will contain all consequences of the privacy preserving answer with respect to u's knowledge base.

For each i where $3 \leq i \leq |\mathbf{U}|+2$ such that $u_{i-2} \neq u$, thus for contexts representing non-querying users, br_i contains one bridge rule $p_1 \leftarrow (j : p_1), \ldots, (j : p_l), \text{not } (i : p_1)$ for $u_j = u$ and $\{p_1, \ldots, p_l\} \in \mathbf{Priv}(u_{i-2})$. The idea is that no belief state can be an equilibrium, in which the querying user derives information which u_{i-2} wants to keep private.

Proposition 1. *Given a* $\mathbf{P3}$ $(\mathbf{MKB}, \mathbf{U}, \mathbf{BK}, \mathbf{Priv}, Q, u)$*, each equilibrium belief state* $(S_1, S_2, S_3, S_4, \ldots, S_{|\mathbf{U}|+3})$ *for* $M_{\mathbf{P3}}$ *is such that* S_2 *is a privacy preserving answer to* $\mathbf{P3}$*. Also, each privacy preserving answer* S *to* $\mathbf{P3}$ *is the second component of an equilibrium for* $M_{\mathbf{P3}}$*.*

Example 3. In the example examined above, consider the $\mathbf{P3}$ $(\mathsf{MedKB}, \{john, jane, acct\}, \mathbf{BK}, \mathbf{Priv}, \mathsf{symptom}(john, X), acct)$. Note that we did not define background knowledge bases for users $john$ and $jane$, but their nature is not important for the example, just assume that they exist. We also have not defined any privacy statement for $acct$, but also this is not important for our example and we will assume that it is empty, that is, $acct$ does not require anything to be kept private. We construct a multi-context system $(C_1, C_2, C_3, C_4, C_5, C_6)$ where $C_1 = (L_{\mathsf{MedKB}}, \mathsf{MedKB}, \emptyset)$, $C_2 = (L_{\mathsf{MedKB}}, \emptyset, br_2)$ with bridge rules br_2 being

$$\mathsf{symptom}(john, s_1) \leftarrow (1 : \mathsf{symptom}(john, s_1)), \mathrm{not}\ (3 : \mathsf{symptom}(john, s_1))$$
$$\mathsf{symptom}(john, s_2) \leftarrow (1 : \mathsf{symptom}(john, s_2)), \mathrm{not}\ (3 : \mathsf{symptom}(john, s_2))$$
$$\mathsf{symptom}(john, s_3) \leftarrow (1 : \mathsf{symptom}(john, s_3)), \mathrm{not}\ (3 : \mathsf{symptom}(john, s_3))$$

then $C_3 = (L_{\mathsf{MedKB}}, \emptyset, br_3)$ with bridge rules br_3 being

$$\mathsf{symptom}(john, s_1) \leftarrow (1 : \mathsf{symptom}(john, s_1)), \mathrm{not}\ (2 : \mathsf{symptom}(john, s_1))$$
$$\mathsf{symptom}(john, s_2) \leftarrow (1 : \mathsf{symptom}(john, s_2)), \mathrm{not}\ (2 : \mathsf{symptom}(john, s_2))$$
$$\mathsf{symptom}(john, s_3) \leftarrow (1 : \mathsf{symptom}(john, s_3)), \mathrm{not}\ (2 : \mathsf{symptom}(john, s_3))$$

then $C_4 = (L_{\mathbf{BK}(john)}, \mathbf{BK}(john), br_4)$ with bridge rules br_4 being

$$\mathsf{disease}(john, cancer) \leftarrow (6 : \mathsf{disease}(john, cancer)), \mathrm{not}\ (4 : \mathsf{disease}(john, cancer))$$

then $C_5 = (L_{\mathbf{BK}(jane)}, \mathbf{BK}(jane), br_5)$ with bridge rules br_5 being

$$\mathsf{disease}(jane, aids) \leftarrow (6 : \mathsf{disease}(jane, aids)), \mathrm{not}\ (5 : \mathsf{disease}(jane, aids)$$

and finally $C_6 = (L_{\mathbf{BK}(acct)}, \mathbf{BK}(acct), br_6)$ with bridge rules br_6 being

$$\mathsf{symptom}(john, s_1) \leftarrow (2 : \mathsf{symptom}(john, s_1))$$
$$\mathsf{symptom}(john, s_2) \leftarrow (2 : \mathsf{symptom}(john, s_2))$$
$$\mathsf{symptom}(john, s_3) \leftarrow (2 : \mathsf{symptom}(john, s_3))$$

$M_{\mathbf{P3}}$ has six equilibria

$$E_1 = (S_{\mathsf{MedKB}}, Ans_1, \mathbf{Ans}(\mathsf{symptom}(john, X)) \setminus Ans_1, Ans_1, \emptyset, \emptyset)$$
$$E_2 = (S_{\mathsf{MedKB}}, Ans_2, \mathbf{Ans}(\mathsf{symptom}(john, X)) \setminus Ans_2, Ans_2, \emptyset, \emptyset)$$
$$E_3 = (S_{\mathsf{MedKB}}, Ans_3, \mathbf{Ans}(\mathsf{symptom}(john, X)) \setminus Ans_3, Ans_3, \emptyset, \emptyset)$$
$$E_4 = (S_{\mathsf{MedKB}}, Ans_4, \mathbf{Ans}(\mathsf{symptom}(john, X)) \setminus Ans_4, Ans_4, \emptyset, \emptyset)$$
$$E_5 = (S_{\mathsf{MedKB}}, Ans_5, \mathbf{Ans}(\mathsf{symptom}(john, X)) \setminus Ans_5, Ans_5, \emptyset, \emptyset)$$
$$E_6 = (S_{\mathsf{MedKB}}, Ans_6, \mathbf{Ans}(\mathsf{symptom}(john, X)) \setminus Ans_6, Ans_6, \emptyset, \emptyset)$$

where S_{MedKB} is as in Example 1 and the second belief set of each E_i is exactly the respective Ans_i of Example 2 and the third belief set is the complement of Ans_i with respect to $\mathbf{Ans}(\mathsf{symptom}(john, X)) = \{\mathsf{symptom}(john, s_1), \mathsf{symptom}(john, s_2), \mathsf{symptom}(john, s_3)\}$.

We would like to point out that in this construction the original knowledge bases are not changed, we only create contexts and bridge rules. All of the background knowledge bases could be multi-context systems themselves; for instance, if the user model for $acct$ foresees that $acct$ is aware of SNOMED and PEPID, then $acct$'s background knowledge base could be a multi-context system comprising these two medical knowledge bases.

In order to obtain maximal privacy preserving answers using the described construction, the simplest way is to postprocessing all privacy preserving answers. More involved solutions would have to interfere with the underlying multi-context system reasoner, for instance by dynamically changing the multi-context system. It is not clear to us at the moment whether it is possible to modify the construction such that the equilibria of the obtained multi-context system correspond directly to the maximal privacy preserving answers.

4 Conclusion and Future Work

We have presented a definition of the privacy preservation problem, which allows for using knowledge bases of different kinds. Finding privacy preserving answers can then be accomplished by building an appropriate multi-context system and computing one of its belief states. Since systems for solving multi-context systems begin to emerge, for example DMCS [1], this also implies that these privacy preserving answers can be effectively computed.

However, usually one is interested in maximal privacy preserving answers. It is unclear to us whether a similar construction as the one presented in this paper can be used for finding privacy preserving answers which are maximal, by just creating appropriate contexts and bridge rules and without modifying the involved knowledge bases or adding new knowledge bases of particular logics. One possible line of investigation would be to examine work on diagnosing inconsistent multi-context systems [5, 2], since in diagnosis tasks there is an implicit minimization criterion, which could be exploited for encoding maximality.

References

1. Bairakdar, S.E., Dao-Tran, M., Eiter, T., Fink, M., Krennwallner, T.: The dmcs solver for distributed nonmonotonic multi-context systems. In: Janhunen, T., Niemelä, I. (eds.) Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA 2010). Lecture Notes in Computer Science, vol. 6341, pp. 352–355. Springer Verlag (2010)
2. Bögl, M., Eiter, T., Fink, M., Schüller, P.: The mcs-ie system for explaining inconsistency in multi-context systems. In: Janhunen, T., Niemelä, I. (eds.) Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA 2010). Lecture Notes in Computer Science, vol. 6341, pp. 356–359. Springer Verlag (2010)
3. Brewka, G., Eiter, T.: Equilibria in heterogeneous nonmonotonic multi-context systems. In: Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI-2007). pp. 385–390. AAAI Press (2007)
4. Dix, J., Faber, W., Subrahmanian, V.: The Relationship between Reasoning about Privacy and Default Logics. In: Sutcliffe, G., Voronkov, A. (eds.) Logic for Programming, Artificial Intelligence, and Reasoning, 12th International Conference, LPAR 2005. Lecture Notes in Computer Science, vol. 3835, pp. 637–650. Springer Verlag (Dec 2005)

5. Eiter, T., Fink, M., Schüller, P., Weinzierl, A.: Finding explanations of inconsistency in multi-context systems. In: Lin, F., Sattler, U., Truszczyński, M. (eds.) Proceedings of the Twelfth International Conference on Knowledge Representation and Reasoning (KR 2010). AAAI Press (2010)
6. Zhao, L., Qian, J., Chang, L., Cai, G.: Using ASP for knowledge management with user authorization. Data & Knowledge Engineering 69(8), 737–762 (2010)

