Privacy-Preserving Query Answering in Logic-based Information Systems

Bernardo Cuenca Grau and **Ian Horrocks** [1]

Abstract. We study privacy guarantees for the owner of an information system who wants to share some of the information in the system with clients while keeping some other information secret. The privacy guarantees ensure that publishing the new information will not compromise the secret one. We present a framework for describing privacy guarantees that generalises existing probabilistic frameworks in relational databases. We also formulate different flavors of privacy-preserving query answering as novel, purely logic-based *reasoning problems* and establish general connections between these reasoning problems and the probabilistic privacy guarantees.

1 Motivation

Privacy protection is an important issue in modern information systems. The digitalization of data on the Web has dramatically increased the risks of private information being either accidentally or maliciously disclosed. These risks have been witnessed by numerous cases of personal data theft from systems that were believed to be secure. The design of information systems that provide *provable* privacy guarantees is, however, still an open problem—in fact, the notion of privacy is itself still open to many interpretations [2].

This paper addresses the problem of *privacy-preserving* query answering. In this setting it is assumed that the information itself is kept secret, but that the owner of the information wants to allow some query access to it while at the same time preventing private information from being revealed. For example, a hospital may want to allow researchers studying prescribing practices to query the patients' records database for information about medicines dispensed in the hospital, but they want to ensure that no information is revealed about the medical conditions of individual patients.

To make this more precise, the hospital wants to check whether answering specified *legal queries* could augment knowledge (from whatever source) that an attacker may have about the answer to a query for patient names and their medical conditions (the so-called *sensitive query*). Taking into account that an attacker may have previous knowledge about the system is of crucial importance, as such knowledge may connect the answers to legal and sensitive queries, and lead to the (partial) revelation of the latter. For example, allowing a query for drugs and the dates on which they were prescribed may seem harmless, but if the attacker knows the dates on which patients have been in hospital and drugs that are used to treat AIDS, then he may deduce that there must be an AIDS patient amongst the group known to be in hospital on a date when AIDS drugs were dispensed.

This problem has been recently investigated in the context of relational databases (DBs) [9, 10, 6]. In these privacy frameworks, the knowledge and/or beliefs about the system of a potential attacker are modeled as a probability distribution over possible states of the information system. Privacy checking then amounts to verifying whether publishing new information, such as the answer to a legal query, could change the probability (from an attacker's perspective) of any particular answer to the sensitive query.

In the first part of this paper, we extend the probabilistic notions of privacy explored in the DB literature to cover a very general class of logic-based languages which includes, for example, ontology languages [12]. Furthermore, since these notions are too strict in practice, we propose ways to weaken them. In the second part, we formulate privacy-preserving query answering in terms of novel, purely logic-based *reasoning problems*. We show that our logic-based notions have natural probabilistic counterparts. Finally, we argue that these reasoning problems are related to existing ones; to illustrate this fact, we point out a connection with the notion of a *conservative extension*, an important concept in modular ontology design [8, 7].

Given the generality of our notion of an information system, we do not make claims concerning computational properties. Our results, however, provide an excellent formal base for studying such properties for particular languages.

2 Logic-based Information Systems

We adopt a general framework for describing logic-based information systems that captures any language whose formal semantics is based on First Order (FO) models; the framework is open toward different mechanisms for selecting admissible models and thus comprises a wide range of languages. We distinguish between *intensional* knowledge (background knowledge about the application domain) and *extensional* knowledge (data involving specific objects of the domain). This allows us to make the usual distinction in KR between schema knowledge and data. The framework here has been adapted from existing general frameworks in the literature [4, 1].

An *Information System Formalism* (ISF) is a tuple $\mathcal{F} = (\Sigma, \mathcal{L}_\mathcal{S}, \mathcal{L}_\mathcal{D}, Sem)$ where Σ is a countably infinite FO-signature, $\mathcal{L}_\mathcal{S}, \mathcal{L}_\mathcal{D}$ are FO-languages over Σ, called the *schema* and *dataset* language respectively, and *Sem* is a specification of the semantics (of which more below). A schema $\mathcal{S}$ (respectively a dataset $\mathcal{D}$) is a set of $\mathcal{L}_\mathcal{S}$-sentences (respectively a set of $\mathcal{L}_\mathcal{D}$-sentences) over Σ.

For example, in relational DBs, Σ is a set of relations and constants; $\mathcal{L}_\mathcal{D}$ only allows for ground atomic formulas, and $\mathcal{L}_\mathcal{S}$ is the language of FO Predicate Logic with equality. Datasets and schemas are called relational instances and relational schemas respectively. In the case of description logic (DL) ontologies, Σ contains unary relations, binary relations and constants; $\mathcal{L}_\mathcal{S}$ is a DL, such as $\mathcal{SHIQ}$ [12], and $\mathcal{L}_\mathcal{D}$ again only allows for ground atomic formulas over the predicates in Σ; Datasets are called ABoxes and schemas TBoxes.

[1] Oxford University Computing Laboratory, UK

The semantics is given by a pair $Sem = (\delta, \circ)$; δ is a function that assigns to each FO-interpretation I over Σ and each possible set $\mathcal{S}$ of $\mathcal{L}_{\mathcal{S}}$-sentences (respectively $\mathcal{L}_{\mathcal{D}}$-sentences $\mathcal{D}$) a truth value $\delta(I, \mathcal{S}) \in \{\mathsf{true}, \mathsf{false}\}$ (respectively $\delta(I, \mathcal{D}) \in \{\mathsf{true}, \mathsf{false}\}$); $\circ$ is a binary operation on sets of interpretations, such that for each pair of sets $\mathcal{M}_1, \mathcal{M}_2$, $\circ$ returns a set of interpretations $\mathcal{M}_3 = \mathcal{M}_1 \circ \mathcal{M}_2$.

An information system (IS) $\mathcal{F}$ is a pair $\mathfrak{I} = (\mathcal{S}, \mathcal{D})$, with $\mathcal{S}$ an $\mathcal{L}_{\mathcal{S}}$-schema, and $\mathcal{D}$ an $\mathcal{L}_{\mathcal{D}}$-dataset. The set of models of $\mathfrak{I}$ is $\mathsf{Mod}(\mathfrak{I}) = \mathsf{Mod}(\mathcal{S}) \circ \mathsf{Mod}(\mathcal{D})$, with $\mathsf{Mod}(\mathcal{S}) = \{I \mid \delta(I, \mathcal{S}) = \mathsf{true}\}$ and $\mathsf{Mod}(\mathcal{D}) = \{I \mid \delta(I, \mathcal{D}) = \mathsf{true}\}$. $\mathfrak{I}$ is satisfiable if $\mathsf{Mod}(\mathfrak{I}) \neq \emptyset$.

For example, in both ontologies and relational DBs, schemas are interpreted in the usual way in FOL: $\delta(I, \mathcal{S}) = \mathsf{true}$ iff $I \models_{FOL} \mathcal{S}$. In $\mathcal{SHIQ}$ ontologies, datasets are also interpreted in the usual way: $\delta(I, \mathcal{D}) = \mathsf{true}$ iff $I \models_{FOL} \mathcal{D}$, and $\circ$ is the intersection between the schema and the dataset models. In relational DBs, however, the data usually has a single model—that is, $\delta(I, \mathcal{D}) = \mathsf{true}$ iff $I = I_{\mathcal{D}}$, where $I_{\mathcal{D}}$ is the minimal Herbrand model of $\mathcal{D}$; The operation $\circ$ is also defined differently: $I_1 \circ I_2 \in \mathsf{Mod}(\mathfrak{I})$ iff $I_2 = I_{\mathcal{D}}$ and $I_{\mathcal{D}} \models_{FOL} \mathcal{S}$.

We are also very permissive w.r.t. query languages. A query language for $\mathcal{F}$ is an FO-language $\mathcal{L}_{Q}$ over Σ. A boolean query Q is an $\mathcal{L}_{Q}$-sentence. The semantics is given by a function $\delta_{\mathcal{L}_Q}$ that assigns to each interpretation I and boolean query Q a truth value $\delta_{\mathcal{L}_Q}(I, Q) \in \{\mathsf{true}, \mathsf{false}\}$. A system $\mathfrak{I}$ entails Q, written $\mathfrak{I} \models_{\mathcal{F}} Q$ if, for each $I \in \mathsf{Mod}(\mathfrak{I})$, $\delta_{\mathcal{L}_Q}(I, Q) = \mathsf{true}$. A general query Q is a $\mathcal{L}_{Q}$-formula, where $\bar{x}$ is the vector of free variables in Q. Let $\sigma_{[\bar{x}/\bar{o}]}$ be a function that, when applied to a general query Q, yields a new boolean query $\sigma_{[\bar{x}/\bar{o}]}(Q)$ by replacing in Q the variables in $\bar{x}$ by the constants in $\bar{o}$. The answer set for Q in $\mathfrak{I}$ is the following set of tuples of constants: $\mathsf{ans}(Q, \mathfrak{I}) = \{\bar{o} \mid \mathfrak{I} \models_{\mathcal{F}} \sigma_{[\bar{x}/\bar{o}]}(Q)\}$.

An example of a query language could be the language of conjunctive queries in both DBs and ontologies. Given a query language $\mathcal{L}_{Q}$, a view over $\mathfrak{I}$ is a pair $\mathcal{V} = (V, \mathbf{v})$, with V—the definition of the view— an $\mathcal{L}_{Q}$-query, and $\mathbf{v}$—the extension of the view— a finite set of tuples of constants, such that $\mathbf{v} = \mathsf{ans}(V, \mathfrak{I})$.

Condition	Set
$[\mathcal{S}\uparrow]$	$\mathsf{Syst}([\mathcal{S}\uparrow]) = \{\mathfrak{I} = (\mathcal{S}', \mathcal{D}) \mid \mathfrak{I} \in \mathbf{IS} \text{ and } \mathcal{S} \subseteq \mathcal{S}'\}$
$[\mathcal{S}^*]$	$\mathsf{Syst}([\mathcal{S}^*]) = \{\mathfrak{I} = (\mathcal{S}, \mathcal{D}) \mid \mathfrak{I} \in \mathbf{IS}\}$
$[\mathbf{V}]$	$\mathsf{Syst}([\mathbf{V}]) = \{\mathfrak{I} \in \mathbf{IS} \mid \text{each } \mathcal{V} \in \mathbf{V} \text{ is a view over } \mathfrak{I}\}$
$[Q = \mathbf{q}]$	$\mathsf{Syst}([Q = \mathbf{q}]) = \{\mathfrak{I} \in \mathbf{IS} \mid \mathsf{ans}(Q, \mathfrak{I}) = \mathbf{q}\}$

Table 1. Conditions on Information Systems

Given $\mathcal{F} = (\Sigma, \mathcal{L}_{\mathcal{S}}, \mathcal{L}_{\mathcal{D}}, Sem)$, we denote by **IS**, **D** the set of all satisfiable systems and datasets respectively in $\mathcal{F}$, and by **Tup** the set of all tuples of constants over Σ. We also consider systems in **IS** that satisfy certain conditions; the conditions we consider are given in Table 1. Given a schema $\mathcal{S}$, the first and second rows in the table represent respectively the set of ISs whose schemas extend $\mathcal{S}$ and are equal to $\mathcal{S}$; given a set of views **V**, the third row represents the set of ISs over which every $\mathcal{V} \in \mathbf{V}$ is a view; finally, given a query Q and an answer set **q**, the last row represents the ISs for which **q** is the answer to Q. We denote with $[C_1, \ldots, C_n]$ the conjunction of conditions $[C_1], \ldots, [C_n]$, and with $\mathsf{Syst}([C_1, \ldots, C_n])$ the subsets of **IS** that satisfy all of $[C_1], \ldots, [C_n]$.

3 The Privacy Problems

Given $\mathcal{F} = (\Sigma, \mathcal{L}_{\mathcal{S}}, \mathcal{L}_{\mathcal{D}}, Sem)$ and a query language $\mathcal{L}_{Q}$, our goal is to study privacy guarantees for Bob —the owner of a system $\mathfrak{I} = (\mathcal{S}, \mathcal{D})$ in **IS**— against the actions of Alice— a potential attacker.

Existing privacy frameworks for DBs[9, 10, 6] assume that the actual data $\mathcal{D}$ is kept hidden. The data to be protected is defined by a query Q, called the *sensitive query*, whose definition is known by Alice. As an external user, Alice can only access the system through a query interface which allows her to ask certain "legal" queries; these legal queries, together with their answers, are represented as a set **V** of views over $\mathfrak{I}$. Bob wants to extend the set of legal queries, i.e., to publish new views. The problem of interest is the following:

The publishing problem: Given $\mathfrak{I} = (\mathcal{S}, \mathcal{D})$, an initial set of views **V** and a final set of views **W** over $\mathfrak{I}$ with $\mathbf{V} \subseteq \mathbf{W}$, verify that no additional information about the answers to Q is disclosed.[2]

$\mathsf{R}(x,y)$	$\mathsf{S}(z,y)$	$\mathsf{T}(z,w,x)$	$\mathsf{F}(z,t)$
$(dis1,drug1)$	$(pat1,drug1)$	$(pat1,male,dis1)$	$(pat1, flo1)$
$(dis2,drug1)$	$(pat2, drug1)$	$(pat2,male,dis2)$	$(pat2, flo2)$
$(dis3, drug2)$	$(pat3, drug2)$	$(pat3, fem, dis3)$	$(pat3, flo3)$
$(dis4, drug2)$	$(pat4, drug2)$	$(pat4, male, dis4)$	$(pat4, flo2)$

Table 2. Example Hidden Dataset

Example 1 The IS of a hospital, modeled in FO-logic, contains data about the following predicates: $\mathsf{R}(x,y)$, which relates diseases to drugs, $\mathsf{S}(z,y)$, which relates patients to their prescribed drugs, $\mathsf{T}(z,w,x)$, which relates patients, their gender, and their diagnosed disease, and $\mathsf{F}(z,t)$ which specifies the floor of the hospital where each patient is located. Their extension in the hidden dataset $\mathcal{D}$ is given in Table 2. The schema $\mathcal{S}$ is public and contains FO-sentences such as $\forall x, y : [\mathsf{R}(x,y) \Rightarrow \mathsf{Disease}(x) \wedge \mathsf{Drug}(y)]$, which ensures that R only relates diseases to drugs, and sentences like $\forall x : [\mathsf{Disease}(x) \Rightarrow \neg\mathsf{Drug}(x))]$, which ensures disjointness between drugs, diseases, patients, genders and floors. $\mathcal{S}$ also models other common-sense knowledge, e.g. that the gender of a patient is unique. Bob does not want to reveal any information about which patients suffer from $dis1$, i.e., the answer to the query $Q(z) = \exists w : [\mathsf{T}(z, w, dis1)]$ should be secret; however, Bob also wants to publish views $\mathcal{V}_1 = (V_1, \mathbf{v}_1)$, and $\mathcal{V}_2 = (V_2, \mathbf{v}_2)$ with $V_1(x,y) \leftarrow \mathsf{F}(z,t)$ and $V_2(z,w) \leftarrow \exists x : [\mathsf{T}(z,w,x)]$, and where $\mathbf{v}_1, \mathbf{v}_2$ are their respective extensions w.r.t. $\mathcal{D}$. Publishing these views could lead to a privacy breach w.r.t. Q. For example, if $\mathcal{S}$ contains a sentence α stating that all the patients in $flo1$ suffer from $dis1$ then, by publishing V_1, Alice could deduce that $pat1$ suffers from $dis1$ and thus belongs to the answer to Q_1, which clearly causes a privacy breach. Even if the identity of patients suffering from $dis1$ is not revealed, the views could still provide useful information to Alice. Suppose that $\mathcal{S}$ contains β stating that $dis1$ is a kind of disease that only affects men; then by publishing $\mathcal{V}_2$ Alice could infer that $pat3$, a woman, cannot be in the answer to Q_1, which would permit Alice to discard possible answers. Such privacy breaches are dataset-dependent: if all patients in $\mathcal{D}$ were male and none of them is on the first floor, then publishing $\mathcal{V}_1$ and $\mathcal{V}_2$ would be harmless. $\diamond$

Existing DB frameworks assume that the schema is *static* and fully *known* by Alice, which are not always reasonable assumptions. For inferential systems like ontologies [12], where the schema participates in query answering by allowing the deduction of new data, Bob may prefer to hide a part of the schema. In fact, some widely used ontologies, such as SNOMED-CT—a component of the Care Record Service in the British Health System—are not fully available. Furthermore, the schema may undergo continuous modifications; indeed many ontologies are updated on a daily basis. To overcome these limitations, we propose to formalise and study the following problems:

The generalised publishing problem: New views or schema axioms are published, but the IS $\mathfrak{I} = (\mathcal{S}, \mathcal{D})$ remains static. Given an initial public schema $\mathcal{S}_1$ and a final public schema $\mathcal{S}_2$ with $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{S}$,

[2] Note that this generalises the "standard" case where $\mathbf{V} = \emptyset$.

initial views $\mathbf{V}$ and final views $\mathbf{W}$ with $\mathbf{V} \subseteq \mathbf{W}$, Bob wants to verify that no additional information about the answers to Q is disclosed.

The system evolution problem: The IS $\mathfrak{I} = (\mathcal{S}, \mathcal{D})$ evolves to $\mathfrak{I}' = (\mathcal{S}', \mathcal{D}')$. Bob wants to ensure that, if it was possible to safely publish certain information before the change, then the same information can be safely published after the change.

DB frameworks are *probabilistic* and apply to the publishing problem [10, 6, 11]. In the next section, we generalise them. Our presentation differs from [10, 6, 11] in two aspects: we consider arbitrary ISFs instead of relational DBs; and we consider the generalised publishing problem: instead of assuming that the schema is fixed and known, we allow for partially secret schemas. We show that known results for DBs can be naturally lifted to our more general setting.

4 Probabilistic Frameworks

The framework by Miklau & Suciu [10] is based on Shannon's information-theoretic notion of *perfect secrecy*. As mentioned before, we present the framework in a more general form.

Alice's (additional) knowledge about the IS being attacked is given as a distribution $P : \mathbf{IS} \rightarrow [0,1]$ over all possible ISs. Given P, the probability that an IS satisfies a condition $[C]$ in Table 1 is as follows: $P([C]) = \sum_{\mathfrak{I} \in \mathsf{Syst}([C])} P(\mathfrak{I})$. Given $[C_1], [C_2], P([C_1] \mid [C_2])$ represents the probability, according to Alice's knowledge, that an IS satisfies $[C_1]$ given that it satisfies $[C_2]$; this can be computed using the Bayes formula: $P([C_1] \mid [C_2]) = \frac{P([C_1, C_2])}{P([C_2])}$

Let $\mathfrak{I} = (\mathcal{S}, \mathcal{D})$ be the system to be protected. Alice initially knows part of the schema $\mathcal{S}_1 \subseteq \mathcal{S}$ and views $\mathbf{V}$ over $\mathfrak{I}$. After publication, she observes the new schema $\mathcal{S}_2$ with $\mathcal{S}_1 \subseteq \mathcal{S}_2$ and views $\mathbf{W} = \mathbf{V} \cup \mathbf{U}$; she is also aware that the real schema $\mathcal{S}$ extends both $\mathcal{S}_1$ and $\mathcal{S}_2$. The a-priori and a-posteriori probabilities, according to Alice's knowledge, that $\mathbf{q}$ is the answer to Q are respectively given as follows:[3]

$$P([Q = \mathbf{q}] \mid [\mathcal{S}_1 \uparrow, \mathbf{V}]) \qquad \text{(a-priori)} \qquad (1)$$

$$P([Q = \mathbf{q}] \mid [\mathcal{S}_2 \uparrow, \mathbf{W}]) \qquad \text{(a-posteriori)} \qquad (2)$$

The privacy condition under consideration is called *perfect privacy*: intuitively, Alice should not learn *anything* about the possible outcomes of Q, whatever her additional knowledge or beliefs (i.e., for *any* P). Note that the condition is trivially satisfied if $\mathcal{S}_1$ and $\mathbf{V}$ already reveal the answer to Q, i.e., if each $\mathfrak{I} \in \mathsf{Syst}([\mathcal{S}_1 \uparrow, \mathbf{V}])$ yields the same outcome to Q; in this case we say that Q is *trivial*.

Example 2 Suppose that in Example 1, the schema $\mathcal{S}$ with $\beta \in \mathcal{S}$ is known, and $\mathcal{V}_2$—the relation between patients and their genders— is published. Suppose that Alice has only vague knowledge about the IS and considers all datasets consistent with $\mathcal{S}$ equally likely. Consider an answer set $\mathbf{q}$ containing *pat*3. Before publishing the view, the probability (1) is non-zero for $\mathbf{q}$, whereas, after publishing $\mathcal{V}_2$, (2) is zero. Intuitively, Alice's knowledge about Q has increased. ◇

Definition 1 (Perfect Privacy). *Perfect privacy* holds if, for each $P : \mathbf{IS} \rightarrow [0,1]$ and $\mathbf{q} \in \mathbf{Tup}$ with (1) well-defined, (2) equals (1).

The framework by Deutsch and Papakonstantinou [6, 11] models Alice's knowledge or beliefs as a distribution $P : \mathbf{Tup} \rightarrow [0,1]$ over the *possible outcomes* of the sensitive query. Here, we present the framework in a more general form.

In Example 1, Alice may believe that the answer to Q is $\mathbf{q}_1 = \{pat1\}$ with $P(\mathbf{q}_1) = 2/3$, $\mathbf{q}_2 = \{pat1, pat2\}$ with $P(\mathbf{q}_2) = 1/6$ and $\mathbf{q}_3 = \{pat1, pat3\}$ with $P(\mathbf{q}_3) = 1/6$. Note the difference with [10], where Alice had prior knowledge about the *possible ISs* themselves.

The distribution P induces possible compatible distributions $P' : \mathbf{IS} \rightarrow [0,1]$ over ISs as follows: P' is compatible with P, written $P' \in \mathsf{Comp}(P)$ if, for each $\mathbf{q}$, the sum of the probabilities of the ISs for which $\mathsf{ans}(Q, \mathfrak{I}) = \mathbf{q}$ is precisely $P(\mathbf{q})$ (i.e., $\sum_{\{\mathfrak{I} \in \mathsf{Syst}([Q=\mathbf{q}])\}} P'(\mathfrak{I}) = P(\mathbf{q})$). Alice's a-priori and a-posteriori knowledge is given respectively by (1) and (2) over P', and the privacy condition is the following:

Definition 2 (Safety). *Safety holds* if, for each $P : \mathbf{Tup} \rightarrow [0,1]$, $P' \in \mathsf{Comp}(P)$, and $\mathbf{q} \in \mathbf{Tup}$ with (1) well-defined, (2) equals (1).

Triviality of Perfect Privacy and Safety: In the relational DB literature, it has been observed that, on the one hand, safety and perfect privacy are closely related [6] and that, on the other hand, they are too strict in practice: revealing any new information, even if apparently irrelevant to Q, causes perfect privacy and safety not to hold—intuitively, this is because the attacker's beliefs can establish a (possibly spurious) connection between *any* revealed information and the answer to the secret query. We show that these results can be naturally lifted to the generalised publishing problem for arbitrary ISFs as follows:

Theorem 1 *For given* $\mathfrak{I}$, Q, $\mathcal{S}_1, \mathcal{S}_2$, *and* $\mathbf{V}$, $\mathbf{W}$*:* (i) *Safety* $\Leftrightarrow$ *Perfect Privacy, and* (ii) *Perfect Privacy* $\Leftrightarrow \mathsf{Syst}([\mathcal{S}_1 \uparrow, \mathbf{V}]) \subseteq \mathsf{Syst}([\mathcal{S}_2 \uparrow, \mathbf{W}])$.

Relaxing Perfect Privacy and Safety: A number of recent papers have tried to weaken these notions. Miklau and Suciu [10] proposed to place constraints on P and consider only product distributions; this amounts to assuming that the tuples in the DB are independent. This assumption, however, is not reasonable if the schema is nontrivial: schema constraints can impose arbitrary correlations between tuples. Other proposals, e.g. [3], involve making (1) only approximately equal to (2). In this paper, we propose two novel notions—quasi-safety and quasi-privacy— that significantly relax Definitions 1 and 2 respectively; we show later on that both notions are equivalent and have a nice logical counterpart in terms of purely logic-based reasoning problems.

Consider the notion of safety. Given $P : \mathbf{Tup} \rightarrow [0,1]$, Definition 2 requires (1) and (2) to coincide for *all* its compatible distributions. Definition 2 can be relaxed by requiring, for each P, only *the existence* of a compatible distribution P' for which (1) and (2) coincide. Moreover, such distribution must be "reasonable" given the public information $\mathcal{S}_1, \mathbf{V}$—that is, if P assigns non-zero probability to $\mathbf{q}_1$, then P' cannot assign zero probability to all ISs that satisfy $[\mathcal{S}_1, \mathbf{V}]$ and yield $\mathbf{q}_1$. Formally, we say that $P' \in \mathsf{Comp}(P)$ is *admissible* for $\mathcal{S}_1, \mathbf{V}$ if, for each $\mathbf{q}$ such that $P(\mathbf{q}) \neq 0$, there is an IS $\mathfrak{I} \in \mathsf{Syst}([\mathcal{S}_1, \mathbf{V}, Q = \mathbf{q}])$ such that $P'(\mathfrak{I}) \neq 0$.

Definition 3 (Quasi-Safety). Quasi-safety holds if, for each $P : \mathbf{Tup} \rightarrow [0,1]$ *there is an* admissible $P' \in \mathsf{Comp}(P)$ s.t., for each $\mathbf{q} \in \mathbf{Tup}$, for which (1) is well-defined, (2) equals (1).

That is, whatever Alice's knowledge or beliefs about the answers to Q, there is always a compatible opinion about the hidden IS that is "reasonable" given the public information and that would not cause her to revise her beliefs after the new information is published. A similar principle can be used for weakening perfect privacy:

Definition 4 (Quasi-Privacy). Quasi-privacy holds if, for each $P : \mathbf{IS} \rightarrow [0,1]$, there is a $P' : \mathbf{IS} \rightarrow [0,1]$ s.t., for each $\mathbf{q} \in \mathbf{Tup}$ for which (1) is well-defined over P, (2) over P' equals (1) over P.

[3] These probabilities are well-defined if $P([\mathcal{S}_1 \uparrow, \mathbf{V}])$ and $P([\mathcal{S}_2 \uparrow, \mathbf{W}])$ are nonzero; that is, if there is a IS with non-zero probability that is compatible with the available information.

That is, whatever Alice's initial beliefs about the hidden IS, she can always revise them such that her opinion about the answers to Q does not change when the new information is published.

5 A Logic-based Framework

In this section, we formalise privacy from a purely logic-based perspective as a guarantee that the published information will not "change the meaning" of the secret query. We propose a collection of privacy conditions that model this notion of meaning change, and consider both the publishing and the evolution problems.

5.1 The Generalised Publishing Problem

The most basic information about Q is obviously its answer. The most dangerous privacy breach occurs when publishing new information reveals part of such answer. In Example 1, before publishing any views, Alice cannot deduce the name of any patient suffering from *dis*1; after publication of $\mathcal{V}_1$, Alice learns that *pat*1 does have *dis*1 and therefore belongs to the answer of Q. We will then say that the set of *certain answers* to Q has changed.

Furthermore, as seen in Example 1, a privacy breach could also occur if Alice can discard possible answers and therefore formulate a "better guess", even if part of the actual answer has not been disclosed. Initially, all possible sets of patients (e.g. $\mathbf{q}_3 = \{pat2, pat3\}$) are possible. Upon publication of $\mathcal{V}_2$, all answers including *pat*3 (e.g. $\mathbf{q}_3 = \{pat2, pat3\}$) become impossible. We will then say that the set of *possible outcomes* of Q has changed.

Possible outcomes and certain answers: Given Q and a condition $[C]$ (see Table 1), the possible outcomes of Q given $[C]$ are as follows:

$$\mathsf{out}([C]) = \{\mathbf{q} \in \mathbf{Tup} \mid \exists \mathfrak{I} \in \mathsf{Syst}([Q = \mathbf{q}, C])\} \qquad (3)$$

The set of *certain answers* of Q given $[C]$ is defined as the common subset of all the possible outcomes: $\mathsf{cert}([C]) = \bigcap \mathsf{out}([C])$.

As argued before, a privacy condition should at least guarantee that the set of certain answers given the initial schema and views stays the same after publishing the new information:[4]

$$\mathsf{cert}([\mathcal{S}_1\!\uparrow, \mathbf{V}]) = \mathsf{cert}([\mathcal{S}_2\!\uparrow, \mathbf{W}]) \qquad (4)$$

A stronger privacy condition can be obtained if we require the set of possible outcomes not to change as follows:

$$\mathsf{out}([\mathcal{S}_1\!\uparrow, \mathbf{V}]) = \mathsf{out}([\mathcal{S}_2\!\uparrow, \mathbf{W}]) \qquad (5)$$

It is ultimately up to the data owner to decide which condition is most appropriate for his application needs.

Monotonicity for answer sets: Sometimes in this section we will focus only on ISFs and query languages that have a *monotonic behavior with respect to answer sets*—that is, if new schema axioms and/or views are published, the set of possible answers to a query Q can only decrease. In the limit, if the whole system is published, then only one answer remains possible, namely the "real" answer for Q against the IS . This property can be formalized as follows:

$$\mathcal{S}_1 \subseteq \mathcal{S}_2 \text{ and } \mathbf{V} \subseteq \mathbf{W} \Rightarrow \mathsf{out}([\mathcal{S}_2^*, \mathbf{W}]) \subseteq \mathsf{out}([\mathcal{S}_1^*, \mathbf{V}]) \qquad (6)$$

Many languages currently used in practice, such as relational DBs and DL ontologies satisfy this property. Checking Condition (5) in ISFs that satisfy Property (6) just requires to consider the initial and final schemas, instead of all their super-sets.

Proposition 1 *If $\mathcal{F}$ satisfies Property (6), then Condition (5) holds iff* $\mathsf{out}([\mathcal{S}_1^*, \mathbf{V}]) \subseteq \mathsf{out}([\mathcal{S}_2^*, \mathbf{W}])$,

In what follows, if a result depends on Property (6), it will be explicitly stated; otherwise, we assume general ISFs and queries.

Bridges between probability and logic: At this stage, we can establish a first general bridge between our logic-based conditions and the probabilistic ones. In particular, it turns out that Condition (5) is equivalent to both quasi-privacy and quasi-safety:

Theorem 2 *Quasi-safety ⇔ Quasi-privacy ⇔ Condition (5).*

Note that Theorem 2, on the one hand, implies that quasi-safety and quasi-privacy are indeed equivalent notions; on the other hand, it provides a natural logical interpretation to our probabilistic weakening of safety and perfect privacy.

Breaches in logic privacy: Condition (5) may still lead to potential security breaches if new schema axioms are published, as shown by the following example:

Example 3 Suppose $\mathcal{L}_\mathcal{S}$ is FO predicate logic, $\mathcal{L}_\mathcal{D}$ only allows for ground atomic formulas, and $\mathcal{L}_Q$ is the language of conjunctive queries. Let A, B be unary predicates and R a binary predicate; consider a Σ' with two constants: a, b. The secret query is $A(x)$. Suppose that Bob publishes $\mathcal{V}_1$ with definition $B(x)$ and extension $\{a, b\}$. Initially, $\mathcal{S}_1 = \emptyset$ and hence all outcomes $\mathbf{Tup} = \{\{\}, \{a\}, \{b\}, \{a, b\}\}$ are possible. Suppose that Bob publishes $\mathcal{S}_2 = \{\forall x : [A(x) \leftrightarrow \exists y : [R(x,y) \wedge B(y)]]\}$. Upon publication of $\mathcal{S}_2$, no possible outcome is ruled out, but $\mathcal{S}_2$ has introduced a *correlation* between $\mathcal{V}_1$ and Q. These correlations could potentially lead to a security breach. ◇

Indeed, even if Alice cannot discard any possible outcome of Q, Bob may want to prevent the new information from establishing potentially dangerous correlations; to this end, we introduce a stronger notion of logic-based privacy.

Strengthening logic privacy: We propose an additional condition in case new schema axioms are published. Our condition is only defined for ISs satisfying Property (6) and it ensures that for each possible dataset $\mathcal{D}$, Alice obtains the same answer for Q independently of whether she considers the initial schema $\mathcal{S}_1$ or the final one $\mathcal{S}_2$. That is, for each $\mathfrak{I} = (\mathcal{S}_2, \mathcal{D}) \in \mathsf{Syst}([\mathcal{S}_2^*, \mathbf{W}])$, the following should hold:

$$\mathsf{ans}(Q, \mathfrak{I}) = \mathsf{ans}(Q, \mathfrak{I}') \qquad (7)$$

where $\mathfrak{I}' = (\mathcal{S}_1, \mathcal{D})$. If we enforce this condition in the example above, we would have that publishing $\mathcal{S}_2$ yields a privacy breach. Indeed, consider $\mathcal{D} = \{R(a,b), B(a), B(b)\}$; we have $\mathsf{ans}(Q, \mathcal{S}_1 = \{\}) = \{\}$, whereas $\mathsf{ans}(Q, \mathcal{S}_2) = \{a\}$. These intuitions motivate the following notion of privacy for ISFs satisfying Property (6):

Definition 5 (Strong Logic-based Privacy). Given $Q, \mathcal{S}_1, \mathcal{S}_2, \mathbf{V}, \mathbf{W}$, strong logic-based privacy holds if Conditions (5) and (7) hold.

The above establishes a middle ground between too strict privacy notions (Definitions 1, 2) and rather permissive ones (Definitions 3, 4). Definition 5 implies that a privacy breach may only occur if the new information correlates the public one to the answers of Q; that is, publishing information that is completely unrelated to Q will not break privacy. Note, however, that if $\mathcal{S}_1 = \mathcal{S}_2$, then Definition 5 reduces to Condition (5) since Condition (7) trivially holds.

A connection with conservative extensions: Definition 5 is close to *conservative extensions*, a well-established notion in mathematical logic, and an important concept in ontology design and reuse [8, 5, 7].

[4] It can be easily seen that Condition (5) implies 4

Conservative extensions have been recently proposed as the basic notion for defining *modules* in ontologies—independent parts of a given theory— and *safe refinements*—extensions of a theory that do not affect certain aspects of the meaning of the original theory. In the context of privacy-preserving query answering, the notion of a *query conservative extension* [7] for monotonic ISFs is of special relevance:

Definition 6 (Query Conservative Extension). [5] Given $\mathcal{S}_1 \subseteq \mathcal{S}_2$, sets $\mathbf{Q}$, $\mathbf{D}$ of queries and datasets respectively, $\mathcal{S}_2$ is a query conservative extension of $\mathcal{S}_1$ w.r.t. $\mathbf{Q}, \mathbf{D}$ if, for each $Q \in \mathbf{Q}$ and $\mathcal{D} \in \mathbf{D}$, we have that $\mathsf{ans}(Q, \Im = (\mathcal{S}_2, \mathcal{D})) = \mathsf{ans}(Q, \Im' = (\mathcal{S}_1, \mathcal{D}))$.

In order to establish a connection between Definitions 5 and 6, let us introduce the following notation. Given $[C]$, we denote the set of datasets that an IS that satisfies $[C]$ can have as follows: $\mathsf{Data}([C]) = \{\mathcal{D} \in \mathbf{D} \mid \exists \Im \in \mathsf{Syst}([C]), \Im$ has dataset $\mathcal{D}\}$.

If $\mathbf{D} = \mathsf{Data}([\mathcal{S}_2^*, \mathbf{W}])$, then Definition 6 corresponds precisely to Condition (7). If $\mathbf{V} = \mathbf{W}$, and $\mathbf{D} = \mathsf{Data}([\mathcal{S}_1^*, \mathbf{V}])$, then Definition 6 is a sufficient condition for strong logic-based privacy.

5.2 The System Evolution Problem

Suppose that the privacy of $\Im = (\mathcal{S}, \mathcal{D})$ w.r.t. a query Q and a set $\mathbf{V}$ of published views has been tested and the system evolves to $\Im' = (\mathcal{S}', \mathcal{D}')$. We want to ensure that $\Im'$ behaves in the same way as $\Im$ w.r.t. the secrecy of Q given $\mathbf{V}$. Such notion of robustness under changes can be characterized as follows. Let $\Im = (\mathcal{S}, \mathcal{D})$, $\Im = (\mathcal{S}', \mathcal{D}')$ be ISs, and let Q be a sensitive query. Consider a notion of security characterized by a predicate $\mathsf{Privacy}(\Im, Q, \mathbf{V})$, e.g. (strong) logic-based privacy, which is evaluated to true if, given the IS $\Im = (\mathcal{S}, \mathcal{D})$, with $\mathcal{S}$ being public, Q is secure for the publication of $\mathbf{V}$.

Definition 7 (Secure Evolution). The evolution of $\Im = (\mathcal{S}, \mathcal{D})$ to $\Im' = (\mathcal{S}', \mathcal{D}')$ is secure w.r.t. Q and $\mathbf{V}$ if $\mathsf{Privacy}(\Im, Q, \mathbf{V})$ implies $\mathsf{Privacy}(\Im', Q, \mathbf{V}')$ with $\mathbf{V}'$ being the views over $\Im'$ with the same view definitions as $\mathbf{V}$.

We distinguish two situations: *(i)* the data changes during the evolution of the system, but the schema remains constant, and *(ii)* the data remains constant, but the schema changes.

Varying the data: We first formulate the notion of *data independence*, which ensures robust evolution w.r.t. changes in the data.

Definition 8 (Data Independence). A notion of privacy is *data-independent* w.r.t. $\mathcal{S}$, Q and $\mathbf{V}$ if, for each $\Im, \Im' \in \mathsf{Syst}([\mathcal{S}^*])$ the evolution of $\Im$ to $\Im'$ is secure w.r.t. Q, $\mathbf{V}$.

It is not hard to see that, given any non-trivial Q and any $\mathcal{S}$, Perfect privacy and safety are data-independent w.r.t. $\mathcal{S}, Q$. In contrast, the notion of privacy derived from Condition 5 is *not* data-independent for all $\mathcal{S}$. Consider Example 1 and suppose that the schema $\mathcal{S}$ contains the sentence β and that the dataset $\mathcal{D}$ only contains male patients. In this case, Condition (5) holds since no possible outcome of Q can be ruled out when publishing $\mathcal{V}_2$; however, if $\mathcal{D}$ evolves to $\mathcal{D}'$ containing a female patient, then the condition is violated. As a consequence, strong logic-based privacy is not data-indepedent and, given Theorem 2, nor are quasi-privacy and quasi-safety.

Data independence for any schema is, indeed, a strict requirement. For ISFs satisfying Property (6), certain schemas and certain views, it is possible to obtain data-independence results:

Proposition 2 *Let $\mathcal{S}$ be a query conservative extension of $\mathcal{S}' = \{\}$ w.r.t. $\mathbf{Q} = \{Q\}$ and $\mathbf{D} = \mathbf{D}$; let $\mathbf{V}$, $\mathbf{V}'$ be s.t. $\mathsf{out}([\mathbf{V}]) = \mathsf{out}([\mathbf{V}'])$. Then (strong) logic-based privacy is data-independent w.r.t. $\mathcal{S}, Q$.*

Proposition 2 guarantees that data independence is obtained for schemas and views that are uncorrelated with the secret query.

Varying the schema: we now assume that the data remains constant and the schema changes. Suppose that, in Example 1, the initial schema $\mathcal{S}$ does not contain β; let $\mathcal{S}' = \mathcal{S} \cup \{\beta\}$ and let the dataset $\mathcal{D}$ contain a female patient. Publishing the names and gender of the patients (view $\mathcal{V}_2$) does not cause a privacy breach since $\mathcal{S}$ does not introduce any correlation between diseases and the gender of patients; however, when $\Im = (\mathcal{S}, \mathcal{D})$ evolves to $\Im' = (\mathcal{S}', \mathcal{D}')$ then such correlation does exist and the publication of $\mathcal{V}_2$ is no longer safe.

Note that, given $Q, \mathcal{D}$, we have that $\mathcal{S}'$ is not a query conservative extension of $\mathcal{S}$. This observation suggests the following sufficient condition for secure evolution of ISFs satisfying Property (6):

Proposition 3 *Let $\mathcal{S}'$ is a query conservative extension of $\mathcal{S}$ w.r.t $\mathbf{Q} = \{Q\}$ and $\mathbf{D} = \mathsf{Data}([\mathcal{S}^*])$; let $\mathsf{out}([\mathcal{S}^*, \mathbf{V}]) = \mathsf{out}([\mathcal{S}^*, \mathbf{V}'])$. Then, the evolution of $\Im = (\mathcal{S}, \mathcal{D})$ to $\Im' = (\mathcal{S}', \mathcal{D})$ is secure w.r.t. $Q, \mathbf{V}$ for both privacy as in Condition (5) and strong logic-based privacy.*

Propositions 2 and 3 establish a bridge between the notions of conservative extension and secure evolution and show that the former can be used to provide sufficient conditions for the latter.

6 Conclusion

In this paper, we have generalised existing results for privacy in databases, and proposed novel privacy conditions. We have proposed a novel logic-based approach and established bridges with existing information-theoretic approaches. Our results provide a deeper fundamental understanding of privacy-preserving query answering and can be used as a starting point for studying the decidability and complexity of the different privacy guarantees for particular languages.

REFERENCES

[1] F. Baader, C. Lutz, H. Sturm, and F. Wolter, 'Fusions of Description Logics and Abstract Description Systems', *JAIR*, **16**, 1–58, (2002).

[2] E. Bertino, S. Jajodia, and P. Samarati, 'Database security: Research and practice', *Inf. Syst.*, **20**(7), 537–556, (1995).

[3] A. Blum, C. Dwork, F. McSherry, and K. Nissim, 'Practical privacy: the sulq framework', in *PODS*, pp. 128–138. ACM, (2005).

[4] D. Calvanese, B. Cuenca Grau, G. De Giacomo, E. Franconi, I. Horrocks, A. Kaplunova, D. Lembo, M. Lenzerini, C. Lutz, D. Martinenghi, R. Moeller, R. Rosati, S. Tessaris, and A.-Y. Turhan. Common framework for representing ontologies. TONES Project Deliverable, 2007.

[5] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler, 'A logical framework for modularity of ontologies', in *IJCAI-07*, pp. 298–304. AAAI, (2007).

[6] A. Deutsch and Y. Papakonstantinou, 'Privacy in database publishing', in *ICDT-2005*, volume 3363 of *LNCS*, pp. 230–245. Springer, (2005).

[7] R. Kontchakov, F. Wolter, and M. Zakharyaschev, 'Modularity in dl lite', in *DL-2007*, (2007).

[8] C. Lutz, D. Walther, and F. Wolter, 'Conservative extensions in expressive description logics', in *IJCAI-07*, pp. 453–459. AAAI, (2007).

[9] A. Machanavajjhala and J. Gehrke, 'On the efficiency of checking perfect privacy', in *PODS-2006*, pp. 163–172. ACM, (2006).

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

[11] A. Nash and A. Deutsch, 'Privacy in GLAV information integration', in *ICDT*, pp. 89–103, (2007).

[12] P.F. Patel-Schneider, P. Hayes, and I. Horrocks. Web ontology language OWL Abstract Syntax and Semantics. W3C Recommendation, 2004.

[13] L. Sweeney, 'K-anoniminity: a model for protecting privacy', *Int. J. on Uncertainty, Fuzziness and Knowledge-based Systems.*, **10**(5), (2002).

[5] In [7], $\mathbf{D}$ and $\mathbf{Q}$ are the sets of all datasets and all queries respectively over a given signature.