Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Logical Foundations of Privacy-Preserving Publishing of Linked Data

Bernardo Cuenca Grau and **Egor V. Kostylev**
Department of Computer Science, University of Oxford, UK

Abstract

The widespread adoption of Linked Data has been driven by the increasing demand for information exchange between organisations, as well as by data publishing regulations in domains such as health care and governance. In this setting, sensitive information is at risk of disclosure since published data can be linked with arbitrary external data sources.

In this paper we lay the foundations of privacy-preserving data publishing (PPDP) in the context of Linked Data. We consider anonymisations of RDF graphs (and, more generally, relational datasets with labelled nulls) and define notions of safe and optimal anonymisations. Safety ensures that the anonymised data can be published with provable protection guarantees against linking attacks, whereas optimality ensures that it preserves as much information from the original data as possible, while satisfying the safety requirement. We establish the complexity of the underpinning decision problems both under open-world semantics inherent to RDF and a closed-world semantics, where we assume that an attacker has complete knowledge over some part of the original data.

1 Introduction

A key advantage of the Linked Data paradigm (Bizer, Heath, and Berners-Lee 2009) is the ability to seamlessly publish and connect semi-structured data on the Web, thus facilitating information sharing and data analysis. Linked Data is based on the RDF data model (Manola and Miller 2004), which provides the means for establishing relationships between objects uniquely identified on the Web, and the RDF query language SPARQL (Harris and Seaborne 2013).

The widespread adoption of Linked Data has been driven by the increasing demand for information exchange between organisations, as well as by regulations in domains such as health care and governance that require certain data to be made available. Data publishing, however, can lead to the disclosure of sensitive information and hence to the violation of individual privacy—a risk that is exacerbated whenever published data can be linked with external data sources.

Privacy-preserving data publishing (PPDP) refers to the problem of protecting individual privacy while at the same time ensuring that published data remains practically useful for analysis. In PPDP there is an emphasis in the publication of actual data; this is in contrast to the less stringent requirements of certain applications, where it suffices to publish the results of data analysis (e.g., statistics about groups of individuals, or association rules), instead of the data itself.

The most popular form of PPDP is *anonymisation*, where explicit individual identifiers and/or the values of certain sensitive attributes are obfuscated. Early approaches to database anonymisation involved the removal of just the identifiers of record owners. Sweeney (2002), however, demonstrated the threats posed by information linkage when they disclosed confidential medical records by linking a medical database where patient names and Social Security Numbers had been anonymised with a public voter list containing postcode, gender, and age information. As a result, PPDP has become an increasingly important problem in recent years and several anonymisation techniques have been proposed in the context of relational databases.[1]

Our goal in this paper is to lay the theoretical foundations for PPDP in the context of Linked Data, with a focus on the semantic requirements that an anonymised RDF graph should satisfy before being released to Web, as well as on the computational complexity of checking whether such requirements are fulfilled. Clearly, these are fundamental steps towards the development of optimised anonymisation algorithms suitable for applications.

In our privacy model, we assume that an anonymised RDF graph G (or, more generally, a relational dataset with labelled null values) is obtained from the original graph G_0 by replacing some occurrences of IRIs in triples with blank nodes. The sensitive information in G_0 that we aim to protect against disclosure is represented by a SPARQL query p, which we refer to as a *policy*. An essential requirement in this setting is that none of the sensitive answers to p hold in G, in which case we say that G is *policy-compliant*. Although policy compliance ensures that the sensitive information is protected when G is considered in isolation, it provides no guarantee against disclosure once G is released to the Web and can be freely linked with arbitrary external data. To address this limitation we formulate an additional *safety* requirement, which ensures that G can be released with provable protection guarantees against linkage attacks.

[1] For details, we refer the reader to the excellent survey in (Fung et al. 2010) and our Related Techniques section.

	Open-world semantics		Closed-world semantics	
	Combined complexity	Data complexity	Combined complexity	Data complexity
Compliance	coNP-c.	in AC^0	Σ_2^p-c.	NP-c.
Safety	Π_2^p-c.	in AC^0	Π_3^p-c.	NP-c.
Optimality	in D_2^p	in AC^0	in D_3^p	in DP

Table 1: Summary of complexity results, where "c." stands for "complete"

Finally, we would like the anonymised graph G to preserve as much information from G_0 as possible while satisfying the aforementioned safety requirement, thus ensuring that the published data remains practically useful; we refer to such most informative anonymisations as *optimal*.

We study the computational complexity of the decision problems underpinning the policy compliance, safety, and optimality requirements. For this, we consider both the *open-world* semantics inherent to RDF and a form of *closed-world* semantics where we assume that an attacker has complete information about certain parts of the original graph G_0. Such closed-world semantics facilitates re-identification of anonymised individuals, thus making the policy compliance and safety requirements more stringent.

Our main technical result is that all the aforementioned reasoning problems are decidable within the the polynomial hierarchy under both open-world and closed-world semantics, with the latter computationally more challenging than the former. Specific complexity results are given in Table 1.

2 Preliminaries

We adopt standard notions in function-free first-order logic with equality. To simplify the presentation we also adopt the unique name assumption (*UNA*), where different constants in formulae cannot be mapped to the same domain element in an interpretation. Dropping the UNA, however, has no effect on any our technical results.

Datasets with Labelled Nulls Let Const and Null be pairwise disjoint sets of *constants* and *(labelled) nulls*, respectively. Assuming a fixed relational vocabulary (i.e., a set of predicate symbols with arities), a *dataset* is a set of atoms with predicates from this vocabulary and terms from $\mathsf{Const} \cup \mathsf{Null}$. A dataset $\mathcal{D}$ is *ground* if it contains no nulls. When talking about logical entailment we view a dataset $\mathcal{D}$ as a sentence $\exists \bar{b} \bigwedge_{\alpha \in \mathcal{D}} \alpha$ where $\bar{b}$ are the nulls occurring in $\mathcal{D}$; clearly, a ground dataset corresponds to a conjunction of ground atoms. According to this interpretation, renamings of nulls preserve logical equivalence; hence, we consider datasets modulo such renamings and assume that $\mathcal{D}_1$ and $\mathcal{D}_2$ have disjoint sets of nulls in use when taking the union $\mathcal{D}_1 \cup \mathcal{D}_2$.

As usual, given datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, a *homomorphism* from $\mathcal{D}_1$ to $\mathcal{D}_2$ is a mapping $h : \mathsf{Const} \cup \mathsf{Null} \to \mathsf{Const} \cup \mathsf{Null}$ such that $h(c) = c$ for each $c \in \mathsf{Const}$ and $h(\mathcal{D}_1) \subseteq \mathcal{D}_2$, where $h(\mathcal{D}_1)$ is the result of applying h to terms in all atoms in $\mathcal{D}_1$. Logical entailment of datasets can be characterised in terms of homomorphisms: $\mathcal{D}_1 \models \mathcal{D}_2$ if and only if there is a homomorphism from $\mathcal{D}_2$ to $\mathcal{D}_1$, for any $\mathcal{D}_1$ and $\mathcal{D}_2$.

Queries A *conjunctive query* (*CQ*) with *free variables* $\bar{x}$ and *existential variables* $\bar{y}$ is a formula $q(\bar{x})$ of the form $\exists \bar{y}\, \varphi(\bar{x}, \bar{y})$, where the *body* $\varphi(\bar{x}, \bar{y})$ is a conjunction of atoms with each term either a constant from Const or a variable from $\bar{x} \cup \bar{y}$. A CQ is *Boolean* if it has no free variables. A tuple of constants $\bar{c}$ from Const is an *answer* to a CQ $q(\bar{x})$ over a dataset $\mathcal{D}$ if $\mathcal{D} \models q(\bar{c})$, where $q(\bar{c})$ is the Boolean query obtained from q by replacing the variables in $\bar{x}$ with the corresponding constants in $\bar{c}$.

RDF and SPARQL All our technical results are stated for general datasets with null values; however, our work is motivated by RDF, so we next define RDF graphs and describe their correspondence to datasets.

Let $\mathbf{I}$, $\mathbf{L}$, and $\mathbf{B}$ be countably infinite pairwise disjoint sets of *IRIs*, *literals*, and *blank nodes*, respectively. An *(RDF) triple* is an element (s, p, o) of $(\mathbf{I} \cup \mathbf{B}) \times \mathbf{I} \times (\mathbf{I} \cup \mathbf{L} \cup \mathbf{B})$, with s called *subject*, p *predicate*, and o *object*. An *RDF graph* is a finite set of triples. RDF comes with a Tarski-style model theory (Hayes 2004), according to which every RDF graph G can be seen as a dataset $\mathcal{D}_G$ over one ternary relation $Triple$ that consists of facts $Triple(s, p, o)$ for each triple (s, p, o) in G, where IRIs $\mathbf{I}$ and literals $\mathbf{L}$ play role of constants, and blank nodes $\mathbf{B}$ play role of nulls; furthermore, blank nodes are local to the graph in which they occur. RDF is equipped with a *merge* operation $G_1 + G_2$ that first renames apart blank nodes in G_1 and G_2 and then constructs the set-theoretic union of their triples; this corresponds precisely to the union of their datasets $\mathcal{D}_{G_1}$ and $\mathcal{D}_{G_2}$.

CQs correspond to the core of the W3C standard query language SPARQL as follows. Given variables $\mathbf{X}$, a basic *SPARQL query* q is of the form SELECT $\bar{x}$ WHERE P, where $\bar{x}$ is a tuple of variables in $\mathbf{X}$ and P is a set of *triple patterns* (s, p, o) with $s, o \in \mathbf{I} \cup \mathbf{L} \cup \mathbf{X}$ and $p \in \mathbf{I} \cup \mathbf{X}$, such that all variables in $\bar{x}$ appear in P. Each such query can be seen as a CQ $q(\bar{x}) = \exists \bar{y} \bigwedge_{(s,p,o) \in P} Triple(s, p, o)$, with $\bar{y}$ the variables in P that are not in $\bar{x}$.

Complexity Classes Many of the computational problems defined in this paper are decidable within the polynomial hierarchy (PH). Recall that the complexity classes in PH are inductively defined in terms of oracle Turing machines as follows: $\Sigma_0^p = \Pi_0^p = \mathrm{P}$, and $\Sigma_{k+1}^p = \mathrm{NP}^{\Sigma_k^p}$, $\Pi_{k+1}^p = \mathrm{coNP}^{\Sigma_k^p}$ for $k > 0$. We also use *difference classes* D_k^p, $k > 0$: a language L is in D_k^p if there are $L_1 \in \Sigma_k^p$ and $L_2 \in \Pi_k^p$ such that $L = L_1 \cap L_2$ (see (Wooldridge and Dunne 2004) for details). This is a generalisation of the class $\mathrm{DP} = \mathrm{D}_1^p$, which consists of those languages that are the intersection of a language in NP and a language in coNP.

3 Logical Framework for PPDP

In this section we present our framework for privacy-preserving data publishing (PPDP) in the context of Linked Data. For the sake of generality, our definitions and theorems are formulated in terms of datasets with nulls; their application to RDF graphs is immediate by the first-order representation and the fact that all our complexity lower bounds can be adapted to hold for a vocabulary with a sin-

gle ternary relation. Our motivating examples are given for RDF graphs.

3.1 Anonymising Linked Data

To illustrate the intuitions behind our approach, let us consider as a running example an excerpt of an RDF graph G_0 representing patient data and consisting of the triples

$$t_1 = (alice, seen\_by, mary),\ t_2 = (bob, seen\_by, mary),\\ t_3 = (mary, dept, oncology).$$

We would like to publish an anonymised version of G_0 while ensuring that the list of patients who have seen an oncologist will not be disclosed, in which case we will say that the anonymisation is *safe*. Additionally, we would like the anonymisation to be *optimal* in the sense that it preserves as much information from G_0 as possible, thus ensuring that the data remains useful in practice.

We will assume that the anonymised graph G is obtained from G_0 by replacing specific occurrences of IRIs in triples with blank nodes. For instance, such G could be obtained by replacing $alice$ in triple t_1, bob in triple t_2 and $mary$ in all three triples with distinct blank nodes b_1, b_2 and b_3, respectively. Semantically, this implies that G is a weakening of the original graph G_0, in the sense that $\mathcal{D}_G$ is homomorphically embeddable into $\mathcal{D}_{G_0}$ and hence $\mathcal{D}_{G_0} \models \mathcal{D}_G$.

Following the mainstream approach in PPDP for databases (e.g., see (Meyerson and Williams 2004)) we formalise anonymisation in terms of *suppressor functions*, which map occurrences of terms in datasets to null values. In contrast to the standard definition, however, we use labelled nulls rather than unlabelled ones.

A *position* s in a dataset $\mathcal{D}$ is a pair $\langle \alpha, j \rangle$ for α an n-ary atom in $\mathcal{D}$ and $1 \leq j \leq n$. Then, the *value* $val(s, \mathcal{D})$ of s in $\mathcal{D}$ is the j-th argument (constant or null) in α.

Definition 1. *Let $\mathcal{D}_0$ be a ground dataset. A $\mathcal{D}_0$-*suppressor *is a function f mapping positions in $\mathcal{D}_0$ to* $\mathsf{Const} \cup \mathsf{Null}$ *such that for all positions s and s' in $\mathcal{D}_0$*
– *if $f(s) \in \mathsf{Const}$ then the value $val(s, \mathcal{D}_0)$ is $f(s)$, and*
– *if $f(s) = f(s')$ then $val(s, \mathcal{D}_0) = val(s', \mathcal{D}_0)$.*
Suppressor f determines the dataset

$$f(\mathcal{D}_0) = \{R(f(\langle \alpha, 1 \rangle), \ldots, f(\langle \alpha, n \rangle)) \mid \\ \alpha \text{ an atom in } \mathcal{D}_0 \text{ over n-ary predicate } R\},$$

which we refer to as an anonymisation.

It is immediate to check that suppressors admit the following characterisation in terms of *strong onto homomorphisms*, i.e., homomorphisms h from $\mathcal{D}_1$ to $\mathcal{D}_2$ with $h(\mathcal{D}_1) = \mathcal{D}_2$.

Proposition 2. *A function f from positions of a ground dataset $\mathcal{D}_0$ to* $\mathsf{Const} \cup \mathsf{Null}$ *is a $\mathcal{D}_0$-suppressor if and only if the multifunction $val(f^{-1}, \mathcal{D}_0)$ is a strong onto homomorphism that maps different atoms to different atoms.*

3.2 Formalising the Sensitive Information

The sensitive information that we aim to protect against disclosure can be naturally represented as a query, which we call a *policy*. For instance, the requirement to protect the list of patients seen by an oncologist can be represented by the following SPARQL query, which has $alice$ and bob as answers over G_0:

$$\mathsf{SELECT}\ x\ \mathsf{WHERE}\ \{(x, seen\_by, y), (y, dept, oncology)\}.$$

A suppressor function f for a dataset $\mathcal{D}_0$ together with a policy p constitute a ***PPDP instance***, as defined next.

Definition 3. *A* PPDP instance *is a triple $(\mathcal{D}_0, f, p)$, where $\mathcal{D}_0$ is a ground dataset, f is a $\mathcal{D}_0$-suppressor, and p is a CQ, called* policy.

3.3 When is Linked Data Publishing Safe?

To protect the sensitive information in our example graph G_0, an essential requirement is that the evaluation of the policy over the anonymised graph does not reveal any of the sensitive answers $alice$ and bob. For instance, a suppressor that replaces all occurrences of $mary$ in G_0 by a single blank node would violate the policy since both sensitive answers follow from the resulting anonymisation. In contrast, by replacing $alice$, bob and $mary$ with blank nodes, as in graph G, we can ensure that no sensitive answer is disclosed.

Definition 4. *A dataset $\mathcal{D}$* complies to *a policy p if $\mathcal{D} \not\models p(\bar{c})$ for any tuple $\bar{c}$ of constants. A PPDP instance $(\mathcal{D}_0, f, p)$ is* policy-compliant *if $f(\mathcal{D}_0)$ complies to p.*

Policy compliance ensures that the sensitive information remains protected when the anonymised data is considered in isolation. It provides, however, no guarantee against disclosure once the anonymised data is released on the Web and can be linked with arbitrary external datasets.

Consider a suppressor that replaces $oncology$ in our example G_0 with a blank node. Although the resulting anonymisation is policy-compliant, the sensitive information can be recovered by linking the anonymised graph with one representing the relationship between doctors and their departments (but saying nothing about patients). We would also run into trouble if we followed the natural approach of replacing $alice$ and bob with blank nodes since the resulting anonymisation could be linked with a graph capturing the relationship between patients and the doctors they saw (but saying nothing about departments).

Therefore, to provide a sensible level of protection against linking attacks we should ensure that the policy is not compromised even if the anonymisation can be freely linked with other graphs. Obviously, anonymising G_0 only makes sense under the assumption that the sensitive information cannot be obtained from external sources only (otherwise, even publishing the empty graph would be problematic); hence, only external graphs complying to the policy are of interest.

Definition 5. *A PPDP instance $(\mathcal{D}_0, f, p)$ is* safe *if, for every dataset $\mathcal{D}'$ complying to p, the union dataset $f(\mathcal{D}_0) \cup \mathcal{D}'$ also complies to p.*

We can ensure safety by replacing all occurrences of $alice$, bob and $mary$ in G_0 with blank nodes b_1, b_2 and b_3, thus obtaining G consisting of triples $(b_1, seen\_by, b_3)$, $(b_2, seen\_by, b_3)$, and $(b_3, dept, oncology)$. Intuitively, graph G is safe because its blank nodes cannot be "accessed" by any external G'; indeed, Definition 5 considers the dataset union, which corresponds to the merge $G + G'$

of G and G' where blank nodes are renamed apart before constructing the set-theoretic union. As a result, if $G + G'$ violates the the policy, then G' alone must introduce $alice$ (or bob) and their explicit connection with the oncology department and hence also violate the policy.

3.4 Safety Under Closed-World Semantics

Definition 5 fits well with the first-order logic semantics of RDF, which is inherently open-world. There are situations, however, when a dataset contains relations for which a smart attacker could easily gather *complete* information about. Consider G'_0 extending G_0 with the following triples:

$$t_4 = (john, spouse, alice), \;\; t_5 = (linda, spouse, bob).$$

We can satisfy the requirements of Definition 5 by replacing $alice$, bob and $mary$ with blank nodes, as we just did in anonymisation G above. However, an attacker having access to a marriage registry database may have complete information about the $spouse$ relation, in which case they can exploit t_4 and t_5 to re-identify b_1 with $alice$ and b_2 with bob.

Such re-identification, however, is only possible under the additional assumption that the marriage registry is complete, that is, no unrecorded marriages can exist. This is in contrast to the open-world setting, where linking the anonymised graph with one containing triples t_4 and t_5 would tell us nothing about the identities of the blank nodes b_1 and b_2.

We can formally represent such closed-world information by means of a CQ corresponding to the SPARQL query SELECT x, y WHERE $(x, spouse, y)$ together with its corresponding answers $(john, alice)$ and $(linda, bob)$ over the original graph G'_0. To ensure that these answers capture all possible triples over the $spouse$ predicate, thus "closing" the $spouse$ predicate, we adapt the standard approach pioneered by Reiter (1992), where a database is seen as a first-order theory with equality axiomatising predicate closure.

Definition 6. *A* closure $[q, Ans]$ *of a CQ* $q(\bar{x}) = \exists \bar{y}\, \varphi(\bar{x}, \bar{y})$ *and set* Ans *of tuples of the same arity is the set of sentences*
- $q(\bar{c})$ *for each* $\bar{c} \in Ans$*;*
- $\forall \bar{x} \forall \bar{y}\, (\varphi(\bar{x}, \bar{y}) \rightarrow \bigvee_{\bar{c} \in Ans} \bar{x} = \bar{c})$*, where* $\bar{x} = \bar{c}$ *stands for* $\bigwedge_{1 \leq i \leq |\bar{x}|} x_i = c_i$ *with* x_i *and* c_i *being the* i*'th components of* $\bar{x}$ *and* $\bar{c}$*.*

In our example, the closure theory fixes the triples with $spouse$ predicate and hence, together with the anonymisation, the attacker can then derive $b_1 = alice$ and $b_2 = bob$.

We can incorporate the notion of closure in our framework by generalising policy compliance and safety as given next.

Definition 7. *A dataset* $\mathcal{D}$ complies *to a policy* p *with respect to a closure* $[q, Ans]$ *if* $\mathcal{D} \cup [q, Ans] \not\models p(\bar{c})$ *for any tuple* $\bar{c}$ *of constants. A PPDP instance* $(\mathcal{D}_0, f, p)$ *is* policy-compliant *with respect to* $[q, Ans]$ *if* $f(\mathcal{D}_0)$ *complies* p *with respect to* $[q, Ans]$.

Since we adopt UNA, a dataset $\mathcal{D}$ can contradict the closure $[q, Ans]$, in which case it does not comply to any policy.

Definition 8. *A PPDP instance* $(\mathcal{D}_0, f, p)$ *is* safe *with respect to a closure* $[q, Ans]$ *if, for each* $\mathcal{D}'$ *complying to* p *with respect to* $[q, Ans]$*, the union* $f(\mathcal{D}_0) \cup \mathcal{D}'$ *also complies to* p *with respect to* $[q, Ans]$.

These notions generalise their open-world counterparts: to capture Definitions 4 and 5 it suffices to consider the closure consisting of the empty Boolean CQ and empty tuple.

3.5 Maximising Data Availability

There is an intrinsic trade-off between privacy preservation and availability of information. Thus, a key challenge is to ensure that the published linked datasets are protected against disclosure of sensitive information while remaining practically useful. We next introduce a notion of *optimality*, which ensures that the published dataset preserves as much information from the original data as possible.

Consider our example G_0 and the (safe) anonymisation G' consisting of $(b_1, seen\_by, b_3)$, $(b_2, seen\_by, b_3)$, and $(b'_3, dept, oncology)$. Such G', however, is not the most informative anonymisation that can be derived from G_0 while ensuring safety. For instance, we could identify b_3 with b'_3 without putting the policy at risk; this yields additional information since we can now conclude that some patient saw an oncologist. To determine which anonymisations are more informative, we introduce a preorder between suppressor functions. Intuitively, a suppressor f_1 is more informative than f_2 if it can be obtained from f_2 by keeping more positions with constants or, as we did in our example, by identifying distinct null values (i.e., RDF blank nodes).

Definition 9. *Given a ground dataset* $\mathcal{D}_0$*, a* $\mathcal{D}_0$*-suppressor* f_1 *is* more specific *than a* $\mathcal{D}_0$*-suppressor* f_2*, written* $f_1 \geq f_2$*, if and only if for all positions* s *and* s' *in* $\mathcal{D}_0$
- *if* $f_2(s) \in \mathsf{Const}$ *then* $f_1(s) = f_2(s)$*, and*
- *if* $f_2(s) = f_2(s')$ *then* $f_1(s) = f_1(s')$.

We write $f_1 > f_2$ *if* $f_1 \geq f_2$*, but* $f_2 \geq f_1$ *does not hold.*

We can now identify as most informative those safe suppressors that are maximal according to the preorder $\geq$.

Definition 10. *A PPDP instance* $(\mathcal{D}_0, f, p)$ *is* optimal *with respect to a closure* $[q, Ans]$ *if*
- *it is safe with respect to* $[q, Ans]$*, and*
- *there is no* f' *such that* $(\mathcal{D}_0, f', p)$ *is safe with respect to* $[q, Ans]$ *and* $f' > f$.

Similarly to Definition 8, optimality under open-world semantics is defined by setting q and Ans as empty.

4 Computational Properties of PPDP

We now study the computational complexity of the reasoning problems underpinning our notions of policy compliance, safety and optimality. In particular, we consider decision problems COMPLIANCE, SAFETY and OPTIMALITY, whose input is, in all cases, a PPDP instance $(\mathcal{D}_0, f, p)$ and a closure $[q, Ans]$, and whose question is as follows.

COMPLIANCE:
Is $(\mathcal{D}_0, f, p)$ policy-compliant with respect to $[q, Ans]$?

SAFETY:
Is $(\mathcal{D}_0, f, p)$ safe with respect to $[q, Ans]$?

OPTIMALITY:
Is $(\mathcal{D}_0, f, p)$ optimal with respect to $[q, Ans]$?

Besides the general form of these problems, when the input is a PPDP instance and a closure, we also study their

data complexity, where both policy and closure are fixed, as well as the intermediate case where only the closure is fixed.

Finally, we consider the *open-world versions* of all these problems that are obtained by setting q and Ans empty.

For simplicity, all the complexity lower bounds in this section are discussed using reductions with no direct encoding in RDF. However, as noted before, all these reductions can be adapted to hold for PPDP instances based on RDF graphs.

4.1 Policy Compliance

We next argue that COMPLIANCE in its general form is Σ_2^p-complete. For membership in Σ_2^p, it suffices to observe that a PPDP instance $(\mathcal{D}_0, f, p)$ is policy-compliant with respect to a closure $[q, Ans]$ if and only if there exists a model of the logical theory $f(\mathcal{D}_0) \cup [q, Ans]$ over which the policy evaluates to the empty set of answers, and if such a model exists, then there exists one of polynomial size. Thus, we can decide COMPLIANCE by first guessing a polynomial-size interpretation and then calling an NP oracle to check whether the interpretation satisfies the required properties.

We obtain a matching Σ_2^p lower bound by reduction of the $\exists\forall$3SAT problem. Given a QBF $\phi = \exists\bar{u}\forall\bar{v}\,\neg\psi(\bar{u}, \bar{v})$ with 3CNF $\psi(\bar{u}, \bar{v})$ over variables $\bar{u} \cup \bar{v}$, we construct $(\mathcal{D}_0, f, p)$ and $[q, Ans]$ such that ϕ is valid if and only if $(\mathcal{D}_0, f, p)$ is policy-compliant with respect to $[q, Ans]$.

The construction uses a vocabulary with unary predicates U, V, a unary predicate Cl_γ for each clause γ in ψ, binary predicates Arg_1, Arg_2 and Arg_3, and a ternary predicate $UVar$. We set $q(x', z') = \exists y'\, UVar(x', y', z')$ and $Ans = \{(c^0, c^1), (c^1, c^0)\}$ for fixed constants c^0 and c^1. (Note that the closure does not depend on QBF ϕ.) The policy p encodes the structure of ϕ as a Boolean CQ with atoms $Cl_\gamma(x_\gamma)$, $Arg_1(x_\gamma, x_{t_1})$, $Arg_2(x_\gamma, x_{t_2})$ and $Arg_3(x_\gamma, x_{t_3})$ for each clause γ over variables t_1, t_2 and t_3 in ψ, an atom $U(x_u)$ for each universal variable $u \in \bar{u}$, and an atom $V(x_v)$ for each existential variable $v \in \bar{v}$. Finally, suppressor f and dataset $\mathcal{D}_0$ are defined such that $f(\mathcal{D}_0)$ consists of the following atoms, where b with super- and subscripts are nulls:

- $Cl_\gamma(b^i_\gamma), Arg_1(b^i_\gamma, b^{s_1}_{t_1}), \ldots, Arg_3(b^i_\gamma, b^{s_3}_{t_3})$ for any clause γ over variables $t_1, \ldots, t_3$ and each satisfying assignment $\{t_1 \mapsto s_1, \ldots, t_3 \mapsto s_3\}$ of γ with number i, $1 \leq i \leq 7$;
- $UVar(b^0_u, b'_u, b^1_u), UVar(b^1_u, b'_u, b^0_u)$ for every $u \in \bar{u}$;
- $V(b^0_v), V(b^1_v)$ for every $v \in \bar{v}$; and $U(c^0)$.

It is then routine to check that ϕ is valid if and only if $f(\mathcal{D}_0)$ complies to p with respect to $[q, Ans]$.

Next, we discuss NP-completeness of COMPLIANCE in data complexity. Membership in NP follows from our Σ_2^p algorithm for the general case since a fixed policy can be evaluated in polynomial time. Hardness can be established by reduction of 3-COLOURABILITY. The idea is similar to the one above: there are exponentially many possibilities to exploit the closure, representing a triangle of colours, and send nulls in $f(\mathcal{D}_0)$ to these colours, that is, there are exponentially many models of $f(\mathcal{D}_0) \cup [q, Ans]$ to check.

Theorem 11. COMPLIANCE *is Σ_2^p-complete, and it remains Σ_2^p-hard even if the closure is fixed. The problem is* NP*-complete in data complexity.*

To conclude this section, we look at COMPLIANCE under open-world semantics, that is, at the case when the closure is empty. This problem is a variation of the standard CQ answering problem in databases since it suffices to check that there are no answers to the policy query p over $f(\mathcal{D}_0)$, that is, the body of p is not homomorphically embeddable into $f(\mathcal{D}_0)$ by mapping the free variables to constants and the existential variables to either constants or nulls.

Theorem 12. *The open-world version of* COMPLIANCE *is* coNP*-complete and in* AC^0 *in data complexity.*

4.2 Safety

We now turn our attention to SAFETY and show that it is Π_3^p-complete in its most general form.

Membership in Π_3^p stems from the key observation that the external datasets $\mathcal{D}'$ we need to consider can be bounded in the size of the policy. Indeed, if there exists a $\mathcal{D}'$ such that $f(\mathcal{D}_0) \cup \mathcal{D}' \cup [q, Ans]$ implies an answer to p, then the image of (the relevant part of) p under the corresponding homomorphism also leads to the disclosure of this answer. Hence, we can decide SAFETY in Π_3^p by considering all $\mathcal{D}'$ of the appropriate size and checking that either *(i)* there is a polynomial-size model of $f(\mathcal{D}_0) \cup \mathcal{D}' \cup [q, Ans]$ that admits no homomorphism from p, or *(ii)* $\mathcal{D}' \cup [q, Ans]$ does not have such a model.

To provide a matching lower bound, we generalise the construction used in Theorem 11. Specifically, we reduce validity of a QBF $\forall\bar{w}\exists\bar{u}\forall\bar{v}\,\neg\psi(\bar{u}, \bar{v}, \bar{w})$ with a 3CNF ψ to SAFETY by following the ideas in Theorem 11 in the encoding of ψ and variables $\bar{u} \cup \bar{v}$, while exploiting the additional variables $\bar{w}$ for checking each relevant external dataset $\mathcal{D}'$.

To establish data complexity bounds, we observe that the aforementioned Π_3^p algorithm yields membership in NP—if policy and closure are fixed, there are only polynomially many $\mathcal{D}'$ to consider, and homomorphism checking becomes feasible in polynomial time. Finally, NP-hardness can be obtained using a reduction similar to that in Theorem 11.

Theorem 13. SAFETY *is Π_3^p-complete, and it remains Π_3^p-hard if the closure is fixed. The problem is* NP*-complete in data complexity.*

Next we look at the open-world version of SAFETY and show that it is of lower complexity, namely Π_2^p-complete.

For the upper bound, it suffices to check that all $\mathcal{D}'$ of relevant (polynomial) size are such that either *(i)* there is no homomorphism from the body of p to $f(\mathcal{D}_0) \cup \mathcal{D}'$ mapping free variables to constants, or *(ii)* there is such homomorphism to $\mathcal{D}'$. Interestingly, an oracle is only needed for *(ii)*.

A matching lower bound is obtained by a reduction of the complement of the *critical tuple problem*, which is known to be Σ_2^p-hard (Miklau and Suciu 2007). This problem is defined as follows: given a ground atom (tuple) α and a Boolean CQ p^*, decide whether α is *critical*, that is, whether there exists a ground dataset $\mathcal{D}^*$ such that the empty tuple is an answer to q over $\mathcal{D}^*$ but not over $\mathcal{D}^* \setminus \{\alpha\}$.[2] For the

[2] In (Miklau and Suciu 2007) the problem is formulated for any CQs, but the CQ in the proof is Boolean.

reduction, we take $\{\alpha\}$ as dataset $\mathcal{D}_0$, suppressor f that trivially maps each position to its value, and p^* as policy p.

Finally, we obtain an AC^0 upper bound for data complexity of SAFETY by showing that it becomes first-order rewritable. Indeed, since p is fixed, we can rewrite all relevant possibilities of $\mathcal{D}'$ together with the policy itself into a fixed first-order logic sentence; checking whether this sentence holds in $f(\mathcal{D}_0)$ is possible in AC^0 (Immerman 1987).

Theorem 14. *The open world version of* SAFETY *is* Π_2^p*-complete and in* AC^0 *in data complexity.*

4.3 Optimality

We conclude this section by providing upper bounds for the different variants of the OPTIMALITY problem. In particular, a decision procedure for each case can be obtained by first checking, using the algorithms in the previous section, that the input instance is safe, and then verifying that none of the (polynomially many) more specific suppressors f' that either identify two nulls in f or map a null to a constant is safe. This procedure shows that OPTIMALITY is in the difference classes D_3^p (the general version) and D_2^p (the open-world version), as well as in DP and in AC^0, respectively, in data complexity.

We hypothesize that these bounds are tight, but leave the investigation of matching lower bounds for future work.

Theorem 15. OPTIMALITY *is in* D_3^p *and in* DP *in data complexity. The open-world version of* OPTIMALITY *is in* D_2^p *and in* AC^0 *in data complexity.*

5 Related Techniques

k-Anonymity The technique that is closest to ours is k-anonymity: a popular method for anonymising databases while providing protection against linkage attacks (Samarati and Sweeney 1998; Sweeney 2002).

The input to the k-anonymity approach is a relational table T; then, some of the entries (i.e., positions) in T are replaced with unnamed nulls so that each tuple t in the anonymised table T' has at least $k-1$ corresponding tuples in T. In this setting, the cost is given by the number of entries in T replaced by nulls, and the goal is to find a k-anonymisation of minimal cost. The underpinning decision problem was shown NP-hard for $k \geq 3$ by Meyerson and Williams (2004) and tractable for $k = 2$ by Blocki and Williams (2010). Practical algorithms were proposed by Bayardo and Agrawal (2005). k-Anonymity has been generalised to handle multiple relations in a database (Nergiz, Clifton, and Nergiz 2009), and to apply only to given sets of attributes in a relation (Wang and Fung 2006). Finally, k-anonymity has also been refined to take into account probabilistic bounds on the attacker's confidence on inferring a sensitive value (Machanavajjhala et al. 2007; Wang and Fung 2006; Wong et al. 2006).

The application of k-anonymity to RDF, where a graph corresponds to a single table with three attributes, is of rather limited use in practice. For instance, the only 2-anonymisation of our example graph G_0 in Section 3 is the trivial one where all IRIs are replaced by fresh nulls. Consequently, our notion of safety provides a much more fine-grained control over the information to be anonymised than k-anonymity since both policies and closed-world requirements can be described by CQs.

Probabilistic techniques In this family of database privacy models the focus is to determine the change in the attacker's probabilistic belief on the sensitive information after accessing the published data; the goal here is to ensure that the information gained by the attacker is small enough. These approaches include (Chawla et al. 2005; Dwork 2006; Rastogi, Hong, and Suciu 2007; Miklau and Suciu 2007).

Graph Anonymisation There has also been considerable recent interest in *graph anonymisation* techniques for social networks, where the goal is to ensure privacy while preserving the global network properties for analysis. Backstrom, Dwork, and Kleinberg (2007), however, showed that the graph's structure can reveal individual identities, even if all node identifiers have been anonymised. To address this problem, Hay et al. (2008) propose the notion of k-candidate anonymity where the requirement is to modify the original anonymised graph via edge additions or deletions until all nodes have the same degree as at least $k-1$ other nodes. Similar notions were studied in (Liu and Terzi 2008; Zhou and Pei 2008). Note, however, that the application of these techniques to RDF is of limited use as they involve anonymising all node identifiers in a graph as a first step.

Privacy for RDF and ontologies The main focus of existing research in the context of RDF and ontologies has been on access control, rather than data publishing and anonymisation. In an access control setting, system's administrators specify by means of declarative access control policies the information accessible to each user. The policy requirements are enforced once the user requests access to information by means of a query. Examples of access control frameworks for linked data and ontologies include (Abel et al. 2007; Bonatti and Sauro 2013; Cuenca Grau et al. 2015; Flouris et al. 2010; Kagal and Pato 2010; Kirrane et al. 2013).

6 Conclusion and Future Work

We have proposed and studied reasoning problems designed to ensure that anonymised RDF graphs can be published on the Semantic Web with provable privacy guarantees.

The problem of RDF anonymisation remains rather unexplored and we see many avenues for future work. First, as noted earlier, we will work on providing tight lower bounds to the complexity of optimality checking. Second, our framework does not yet capture OWL 2 ontologies, which are used in many applications to enrich the semantics of RDF graphs. We anticipate that the introduction of ontologies into the picture will lead to significant technical challenges, especially in combination with closed-world semantics; an interesting starting point to address these challenges is the recent work by Ngo, Ortiz, and Simkus (2015) and Seylan, Franconi, and de Bruijn (2009). Finally, and most importantly, our decidability results open the door to the future design of practical anonymisation algorithms.

Acknowledgements

This work was supported by the Royal Society under a University Research Fellowship and by the EPSRC projects Score!, MaSI^3, and DBOnto.

References

Abel, F.; Coi, J. L. D.; Henze, N.; Koesling, A. W.; Krause, D.; and Olmedilla, D. 2007. Enabling advanced and context-dependent access control in RDF stores. In *ISWC*, 1–14.

Backstrom, L.; Dwork, C.; and Kleinberg, J. M. 2007. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW*, 181–190.

Bayardo, R., and Agrawal, R. 2005. Data privacy through optimal k-anonymization. In *ICDE*, 217–228.

Bizer, C.; Heath, T.; and Berners-Lee, T. 2009. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.* 5(3):1–22.

Blocki, J., and Williams, R. 2010. Resolving the complexity of some data privacy problems. In *ICALP*, 393–404.

Bonatti, P. A., and Sauro, L. 2013. A confidentiality model for ontologies. In *ISWC*, 17–32.

Chawla, S.; Dwork, C.; McSherry, F.; Smith, A.; and Wee, H. 2005. Toward privacy in public databases. In *TCC*, 363–385.

Cuenca Grau, B.; Kharlamov, E.; Kostylev, E. V.; and Zheleznyakov, D. 2015. Controlled query evaluation for datalog and OWL 2 profile ontologies. In *IJCAI*, 2883–2889.

Dwork, C. 2006. Differential privacy. In *ICALP*, 1–12.

Flouris, G.; Fundulaki, I.; Michou, M.; and Antoniou, G. 2010. Controlling access to RDF graphs. In *FIS*, 107–117.

Fung, B. C. M.; Wang, K.; Chen, R.; and Yu, P. S. 2010. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys* 42(4):14:1–14:53.

Harris, S., and Seaborne, A. 2013. SPARQL 1.1 Query language. W3C Recommendation.

Hay, M.; Miklau, G.; Jensen, D.; Towsley, D. F.; and Weis, P. 2008. Resisting structural re-identification in anonymized social networks. *PVLDB* 1(1):102–114.

Hayes, P. 2004. RDF Semantics. W3C Recommendation.

Immerman, N. 1987. Expressibility as a complexity measure: results and directions. In *Proceedings of the Second Conference on Structure in Complexity Theory.*

Kagal, L., and Pato, J. 2010. Preserving privacy based on semantic policy tools. *IEEE Security & Privacy* 8(4):25–30.

Kirrane, S.; Abdelrahman, A.; Mileo, A.; and Decker, S. 2013. Secure manipulation of linked data. In *ISWC*, 248–263.

Liu, K., and Terzi, E. 2008. Towards identity anonymization on graphs. In *SIGMOD*, 93–106.

Machanavajjhala, A.; Kifer, D.; Gehrke, J.; and Venkitasubramaniam, M. 2007. *L*-diversity: Privacy beyond *k*-anonymity. *TKDD* 1(1).

Manola, F., and Miller, E. 2004. RDF Primer. W3C Recommendation.

Meyerson, A., and Williams, R. 2004. On the complexity of optimal k-anonymity. In *PODS*, 223–228.

Miklau, G., and Suciu, D. 2007. A Formal Analysis of Information Disclosure in Data Exchange. *J. Comput. Syst. Sci.* 73(3):507–534.

Nergiz, M. E.; Clifton, C.; and Nergiz, A. E. 2009. Multirelational k-anonymity. *IEEE Trans. Knowl. Data Eng.* 21(8):1104–1117.

Ngo, N.; Ortiz, M.; and Simkus, M. 2015. The combined complexity of reasoning with closed predicates in description logics. In *DL*.

Rastogi, V.; Hong, S.; and Suciu, D. 2007. The boundary between privacy and utility in data publishing. In *VLDB*, 531–542.

Reiter, R. 1992. What should a database know? *J. Log. Program.* 14(1&2):127–153.

Samarati, P., and Sweeney, L. 1998. Generalizing data to provide anonymity when disclosing information (abstract). In *PODS*, 188.

Seylan, I.; Franconi, E.; and de Bruijn, J. 2009. Effective query rewriting with ontologies over dboxes. In *IJCAI*, 923–925.

Sweeney, L. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10(5):557–570.

Wang, K., and Fung, B. C. M. 2006. Anonymizing sequential releases. In *SIGKDD*, 414–423.

Wong, R. C.; Li, J.; Fu, A. W.; and Wang, K. 2006. (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *SIGKDD*, 754–759.

Wooldridge, M., and Dunne, P. E. 2004. On the computational complexity of qualitative coalitional games. *Artificial Intelligence* 158(1):27–73.

Zhou, B., and Pei, J. 2008. Preserving privacy in social networks against neighborhood attacks. In *ICDE*, 506–515.