Lost in the Crowd: k-unmatchability in Anonymized Knowledge Graphs

Piero Andrea Bonatti[1] , **Francesco Magliocca**[1] , **Luigi Sauro**[1]
[1]University of Naples Federico II
{pab, francesco.magliocca, luigi.sauro}@unina.it

Abstract

This paper introduces and investigates *k-unmatchability*, a counterpart of k-anonymity for knowledge graphs. Like k-anonymity, k-unmatchability enhances privacy by ensuring that any individual in any external source can always be matched to either none or at least k different anonymized individuals. The tradeoff between privacy protection and information loss can be controlled with parameter k.sk We analyze the data complexity of k-unmatchability under different notions of anonymization.

1 Introduction

The Semantic Web paradigms excel in connecting diverse data, which increasingly involve personal, possibly sensitive information such as medical records, governance or financial data (Bizer 2009; Bizer, Heath, and Berners-Lee 2009). The confidentiality of this information shall be protected to meet application requirements and comply with personal data protection regulations (Bonatti and Sauro 2013; Cuenca Grau and Kostylev 2019). The IRIs adopted by the W3C standard RDF uniquely identify resources on the Web, thereby facilitating the linkage of different pieces of information related to the same person. Such resource linkage is a major source of citizen profiling and privacy breaches.

Privacy-related risks can be mitigated by anonymizing RDF graphs. Anonymization can be achieved by substituting *blank nodes* (*blanks*, for short) for nodes such as the IRIs of persons and the literals that denote explicit identifiers, like name and SSN (Cuenca Grau and Kostylev 2019). Such *suppression* operations can break the explicit connections between the knowledge encoded in (multiple) RDF graphs and the persons it refers to; suppression is analogous to explicit identifier removal in database anonymization.

Unfortunately, removing explicit identifiers does not suffice to protect privacy. Latanya Sweeney proved this by re-identifying the governor of Massachusetts in an "anonymized" medical database. We briefly recall her attack – as reported in (Ohm 2010) – which will be helpful in understanding our approach. Sweeney first leveraged the common knowledge that the governor of Massachusetts had collapsed unconscious during a public event, and was taken to the Deaconess Waltham Hospital. Thus, it was certain that the governor was represented in the hospital's public database of patient data (where all explicit identifiers had been removed). The governor was also listed in Massachusetts' public database of voters, where the governor's name was associated to his postcode, birthdate, and gender. A join of the two public databases on these three attributes produced one match, that associated the governor's name to his diagnosis (a severe data breach). The uniqueness of the match was not by chance, as postcode, birthdate, and gender constitute a *quasi identifier*, that is, a group of attributes that uniquely identify a person with very high probability. The statistical analysis in (Sweeney 2000) showed that the above quasi identifier uniquely identified all Massachusetts' citizens of age 25–34 and $\geq$45, while in general it identified 98% of the entire population.

After this proof-of-concept attack, various anonymization techniques have been proposed to strengthen privacy-preserving data publishing and counteract information linkage (Dwork 2006; Fung et al. 2010). In particular, *k-anonymity* aims at *preventing re-identification* by ensuring that each record in a dataset is indistinguishable from at least $k-1$ other records (Sweeney 2002; di Vimercati et al. 2023; Bayardo and Agrawal 2005; Machanavajjhala et al. 2007). The uncertainty of re-identification grows with parameter k, as does the amount of concealed information. Implementations of k-anonymity are available for relational databases, yet no exact counterpart has been devised for knowledge graphs, as commonly used in the Semantic Web. Moreover, the complexity of anonymization has not yet been systematically studied in this context.

We address this gap by introducing and investigating *k-unmatchability*, a counterpart of k-anonymity for knowledge graphs. We carry out our analysis in a framework similar to (Cuenca Grau and Kostylev 2019), where the original confidentiality criteria, based on the non-derivability of secret formulae (which cannot mitigate Sweeney's attack), are replaced by k-unmatchability, which is grounded on the undistinguishability of individuals to prevent re-identification.

We examine three types of anonymizations: strict, equipotent, and poly anonymizations. The first two types are tightly related to the suppressors of (Cuenca Grau and Kostylev 2019), which substitute blanks for constants (e.g. IRIs or literals). For each anonymization type, we consider three decision problems, namely checking whether (i) a specific anonymization is k-unmatchable; (ii) a k-unmatchable anonymization exists; (iii) a k-unmatchable anonymization

exists whose cost is bound by a threshold l. Our results almost completely characterize the above decision problems in terms of data complexity, i.e. when the redundancy parameter k is fixed.

Section 2 is devoted to preliminaries. Section 3 introduces the anonymization framework, along with an illustrative example. The data complexity of k-unmatchability is analyzed in Section 4. Section 5 offers an overview of related work. Section 6 concludes the paper with some final remarks and an outline of future research directions.

2 Preliminaries

2.1 Complexity classes

K-unmatchability is related to the *graph isomorphism* and *subgraph isomorphism* problems. The former consists in deciding whether two graphs are isomorphic. The latter, given two graphs G and H, consists in deciding whether some subgraph of G is isomorphic to H. The subgraph isomorphism problem is in the class NPc of NP-complete problems, while the graph isomorphism problem is conjectured to be in $\mathrm{NP} \setminus (\mathrm{NPc} \cup \mathrm{P})$. We say that L is *gi-hard* if *graph isomorphism* is many-one reducible to L in polynomial time. GI denotes the class of problems L that are polynomial-time Turing reducible to *graph isomorphism*.

2.2 Knowledge graphs

The formalism we deal with in this paper is based on the following denumerable sets of symbols: $\mathsf{N_C}$ (concept names), $\mathsf{N_R}$ (role names), $\mathsf{N_I}$ (individual names) and a subset $\mathsf{N_B} \subseteq \mathsf{N_I}$ of "blank" names. Letters A, B (possibly with subscripts and superscripts) denote members of $\mathsf{N_C}$, letters P, R denote members of $\mathsf{N_R}$, letter a ranges over all elements in $\mathsf{N_I}$, while letters c, d denote members of $\mathsf{N_I} \setminus \mathsf{N_B}$ (a.k.a. constant names), and letter b denotes blanks in $\mathsf{N_B}$.

An *ABox* $\mathcal{A}$ is a finite set of axioms of the form $A(a)$ and $P(a_1, a_2)$. Hereafter, $sig(\mathcal{A})$ denotes the set of individual names occurring in $\mathcal{A}$.

Remark 1. Any ABox $\mathcal{A}$ can be represented as an RDF graph, where an assertion $P(a_1, a_2)$ corresponds to a triple (a_1, P, a_2), and assertions of the form $A(a)$ can be translated to $(a, \mathsf{rdf{:}type}, A)$. In RDF graphs, literals (e.g. strings and numbers) may be used as attribute values. For our purposes, it is inconsequential whether an individual is an IRI or a literal; we treat both as constant names. Note that not all RDF graphs can be represented as ABoxes, as RDF lacks strict typing distinction between individuals and predicates. Here we deal with first-order graphs only. □

2.3 Anonymizations

In (Cuenca Grau and Kostylev 2019), anonymized ABoxes are produced by *suppressors*, which (non-uniformly) substitute blanks for constant *occurrences*. We shall rather work with the inverse of suppressors, that we call *re-identifications* (which *uniformly* map blanks back to the corresponding concealed values) because they are simpler to define and handle, and suffice for our purposes. Re-identifications are special cases of substitutions:

A *substitution* over an ABox $\mathcal{A}$ is a function $\tau : sig(\mathcal{A}) \to \mathsf{N_I}$ such that if $c \in \mathsf{N_I} \setminus \mathsf{N_B}$ then $\tau(c) = c$.[1] Given a substitution τ over an ABox $\mathcal{A}$, $\tau(\mathcal{A})$ denotes the ABox obtained from $\mathcal{A}$ by uniformly replacing each $a \in sig(\mathcal{A})$ with $\tau(a)$. Then, given two ABoxes $\mathcal{A}_1$ and $\mathcal{A}_2$, we say that a substitution τ over $\mathcal{A}_1$ is a *re-identification* of $\mathcal{A}_1$ in $\mathcal{A}_2$ if $\tau(\mathcal{A}_1) = \mathcal{A}_2$. Hereafter, for the sake of brevity, we omit from the specifications of substitutions the individuals that are mapped on themselves.

We are now prepared to introduce the notion of anonymization. In general, an ABox can be anonymized by turning some occurrences of its constants into blanks. Different occurrences of the same constant may be mapped to different blanks in order to disconnect parts of the RDF graphs. It is also possible to transform a single axiom into multiple copies thereof (each of which replaces the axiom's constants with different blanks), thereby increasing the number of assertions; the purpose is increasing redundancy within the ABox to obstacle the process of matching blanks back to known individuals. It is prohibited to transform different individuals into the same blank (anonymization should not lie by asserting that two individuals are the same when they are not). Note that $\mathcal{A}_1$ is an anonymization of $\mathcal{A}_2$ obtained with such transformations if and only if there exists a re-identification that allows the reconstruction of $\mathcal{A}_2$ from $\mathcal{A}_1$ (by mapping the blanks of $\mathcal{A}_1$ back to their original value). The next definition introduces a taxonomy comprising three distinct types of anonymization, depending on which of the above transformations are allowed.

Definition 2. *Let $\mathcal{A}_1, \mathcal{A}_2$ be two ABoxes, we say that*

- *$\mathcal{A}_1$ is a* poly anonymization *of $\mathcal{A}_2$ if there exists a re-identification τ of $\mathcal{A}_1$ in $\mathcal{A}_2$.*
- *$\mathcal{A}_1$ is an* equipotent anonymization *of $\mathcal{A}_2$ if $\mathcal{A}_1$ is a poly anonymization of $\mathcal{A}_2$ and $|\mathcal{A}_1| = |\mathcal{A}_2|$.*
- *$\mathcal{A}_1$ is a* strict anonymization *of $\mathcal{A}_2$ if $\mathcal{A}_1$ is an equipotent anonymization of $\mathcal{A}_2$ through an injective re-indentification τ.*

The above anonymization types are ordered by increasing information loss. Poly anonymizations are the most general kind of anonymizations. As their name suggests, they may create multiple anonymized copies of some axioms, and, consequently, inflate the ABox. In equipotent anonymizations, the original number of assertions shall be preserved, so duplication of facts is forbidden; however, different occurrences of the same individual may still be replaced with distinct blank nodes (this corresponds to deleting some equalities between triple arguments). Both of these approaches may affect the result of counting queries, so they are unsuitable for some applications, e.g. some statistical analyses on the data. In contrast, strict anonymizations uniformly replace constant names with blanks (and, consequently, preserve statistics).

Remark 3. It is worth noting that $\mathcal{A}_1$ is an equipotent anonymization of $\mathcal{A}_2$ if, and only if, $\mathcal{A}_1 = f(\mathcal{A}_2)$, for

[1] Since $\mathsf{N_B} \subseteq \mathsf{N_I}$, a substitution can potentially map a blank to another blank, possibly itself.

some suppressor f of $\mathcal{A}_2$, as defined in (Cuenca Grau and Kostylev 2019).

Moreover, if $\mathcal{A}_1$ is a strict anonymization of $\mathcal{A}_2$ through the re-identification τ, then the inverse of τ corresponds to a *strict suppressor*.[2] □

When possible, information loss should be avoided, to retain knowledge utility. Thus it is interesting to study the properties of each of the above anonymization types.

3 The Anonymization Framework

In what follows, definitions and decision problems will be parameterized according to which type of anonymization is used. To ease readability, we use $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$ as a prefix to specify anonymization types.

Our framework consists of a tuple $\langle \mathcal{A}, \mathcal{Q}, x, \mathcal{A}_p, \mathcal{A}_a \rangle$ where:

- $\mathcal{A}$ is the original ABox, not accessible to the attacker. In Sweeney's attack scenario, $\mathcal{A}$ is the full medical database, with all explicit identifiers.
- $\mathcal{Q} \subseteq \mathsf{N_I} \setminus \mathsf{N_B}$ is a nonempty, finite set of constant names called *subjects* that occur in $\mathcal{A}$. $\mathcal{Q}$ specifies which constants shall be safeguarded against high-probability identification. The set $\mathcal{Q}$ typically comprises individuals from particular classes, such as Person or Patient.
- $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$ indicates the specific type of anonymization applied. We make the assumption that x is accessible to potential attackers.
- $\mathcal{A}_p$ is a public x-anonymization of $\mathcal{A}$.
- $\mathcal{A}_a$ is a subset of $\mathcal{A}$, up to blank renaming, that represents the attacker's knowledge about the contents of $\mathcal{A}$. For instance, in Sweeney's attack, $\mathcal{A}_a$ encodes the information about the governor obtained from the voter list (name, birthdate, postcode, and gender) which is surely contained in the hospital's database, too.

Hereafter $\mathcal{A}$, $\mathcal{Q}$, x, $\mathcal{A}_p$, and $\mathcal{A}_a$ refer to an arbitrary but fixed anonymization framework.

Our confidentiality criterion requires a notion of matching, which represents a plausible way – to the eyes of the attacker – to match her knowledge $\mathcal{A}_a$ to a subset $\mathcal{A}'_p$ of $\mathcal{A}_p$.

Definition 4. *We say that a substitution τ over $\mathcal{A}_p$ is an* x-matching *of $\mathcal{A}_p$ in $\mathcal{A}_a$ (with $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$) if there exists a subset $\mathcal{A}'_p \subseteq \mathcal{A}_p$ such that $\mathcal{A}'_p$ is an x-anonymization of $\mathcal{A}_a$ through τ.*

Hereafter, $T_x[\mathcal{A}_p, \mathcal{A}_a]$, will denote the set of all x-matchings of $\mathcal{A}_p$ in $\mathcal{A}_a$. Moreover, for all pairs of individual names $a_1 \in sig(\mathcal{A}_p)$, $a_2 \in sig(\mathcal{A}_a)$, and $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$, we say that a_1 is *x-matchable in a_2 with $T_x[\mathcal{A}_p, \mathcal{A}_a]$*, if $\tau(a_1) = a_2$ holds for some $\tau \in T_x[\mathcal{A}_p, \mathcal{A}_a]$. $T_x[\mathcal{A}_p, \mathcal{A}_a]$ will be omitted when clear from context.

Next, we introduce the target confidentiality criterion (the goal of anonymization), namely, k-unmatchability. The purpose of re-identification attacks (generalizing Sweeney's scenario) is identifying, for at least one subject $c \in \mathcal{Q}$ occurring in $\mathcal{A}_a$, a small set S of blanks from $\mathcal{A}_p$ (where "small" means $|S| < k$) such that c surely "corresponds" to one of the blanks in S. The following definition of k-unmatchability states that the above attack cannot be attained, using matchings as a formalization of "correspondence".

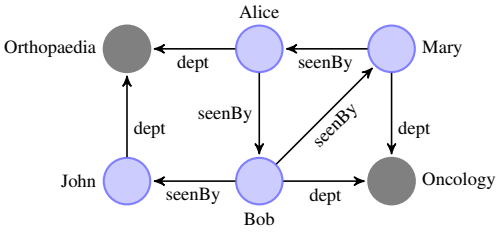

Figure 1: The Abox $\mathcal{A}$ in Example 6.

Definition 5 (k-x-unmatchable)**.** *We say that $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}_a$ w.r.t. $\mathcal{Q}$ iff*

(i) $sig(\mathcal{A}_p) \cap \mathcal{Q} = \emptyset$, *and*

(ii) there exist no individuals $c \in \mathcal{Q}$ and $b_1, \ldots, b_{k-1} \in sig(\mathcal{A}_p) \cap \mathsf{N_B}$ such that for all $\tau \in T_x[\mathcal{A}_p, \mathcal{A}_a]$, $c \in \{\tau(b_1), \ldots, \tau(b_{k-1})\}$.

Note that $\mathcal{A}_p$ and τ have the same type x. This models the assumption that the attacker knows the anonymization type x and – accordingly – considers only x-matchings while deanonymizing the blanks of $\mathcal{A}_p$.

The next example, inspired by (Cuenca Grau and Kostylev 2019), illustrates a scenario of 3-unmatchability where it is not necessary to anonymize all person identifiers.

Example 6. Consider an ABox $\mathcal{A}$ representing patient data as depicted in Figure 1, and an attacker whose ABox $\mathcal{A}_a$ consists of the assertions:

$$\mathsf{seenBy}(\mathsf{Alice}, \mathsf{Bob}), \qquad \mathsf{seenBy}(\mathsf{Bob}, \mathsf{Mary});$$

The patients who have been seen by an oncologist should not be identifiable by the attacker, consequently $\mathcal{Q}$ consists of Bob and Alice. Then, we publish a strict-anonymization $\mathcal{A}_p$ of $\mathcal{A}$ where Bob, Mary, and Alice are transformed into blank nodes b_1, b_2, and b_3, respectively. Clearly, there exists a strict-matching τ_1 of $\mathcal{A}_p$ in $\mathcal{A}_a$ where b_1 is mapped into Bob, b_2 into Mary, and b_3 into Alice. However, by symmetry, there also exist two other matchings τ_2 and τ_3 obtained by rotating the previous matching – i.e., b_2 is mapped into Bob (b_3 into Mary, and b_1 into Alice), and b_3 is mapped into Bob (b_1 into Mary, and b_2 into Alice), respectively. Consequently, $\mathcal{A}_p$ is 3-strict-unmatchable in $\mathcal{A}_a$.

Note that (i) John needs not be anonymized; (ii) if Mary were not anonymized, then the attacker could reconstruct the original ABox $\mathcal{A}$; (iii) if $\mathcal{A}_a$ contained also the assertion $\mathsf{dept}(\mathsf{Bob}, \mathsf{Oncology})$, then $\mathcal{A}_p$ would only be 2-unmatchable in $\mathcal{A}_a$ (since τ_2 where b_3 is mapped into Mary is not a strict-matching anymore). □

As it should be expected, k-unmatchability is monotonic with respect to the attacker's ABox.

[2] To accommodate page limits, comprehensive proofs will be provided in an extended version of this paper.

Theorem 7. *Given a public ABox $\mathcal{A}_p$ and two ABoxes $\mathcal{A}'_a$ and $\mathcal{A}_a$ such that $\mathcal{A}'_a \subseteq \mathcal{A}_a$, if $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}_a$ w.r.t. $\mathcal{Q}$, then $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}'_a$ w.r.t. $\mathcal{Q}$, for all $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$.*

Proof. By assumption $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}_a$ w.r.t. $\mathcal{Q}$ and so $sig(\mathcal{A}_p) \cap \mathcal{Q} = \emptyset$. Now, let $c \in \mathcal{Q}$ and $b_1, \ldots, b_{k-1} \in sig(\mathcal{A}_p) \cap \mathsf{N_B}$; again by the assumption that $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}_a$ w.r.t. $\mathcal{Q}$, we get that there exists $\tau \in T_x[\mathcal{A}_p, \mathcal{A}_a]$ such that $c \notin \{\tau(b_1), \ldots, \tau(b_{k-1})\}$. Finally, since $\mathcal{A}'_a \subseteq \mathcal{A}_a$, then $\tau \in T_x[\mathcal{A}_p, \mathcal{A}'_a]$ too, so we deduce that $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}'_a$ w.r.t. $\mathcal{Q}$. □

As a corollary, when $\mathcal{A}_a$ cannot be reliably estimated, the pessimistic assumption $\mathcal{A}_a = \mathcal{A}$ constitutes a safe option.

Corollary 8. *If $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$ then for all $\mathcal{A}_a \subseteq \mathcal{A}$, $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}_a$ w.r.t. $\mathcal{Q}$ (and viceversa).*

We conclude this section showing that k-unmatchability is preserved by "chained" anonymizations.

Theorem 9. *Let $\mathcal{A}$ be an ABox, $\mathcal{A}_a \subseteq \mathcal{A}$ and $\mathcal{Q}$ be a set of subjects in $\mathcal{A}$. Let $\mathcal{A}_p$ be an x-anonymization of $\mathcal{A}$ and $\bar{\mathcal{A}}_p$ an x-anonymization of $\mathcal{A}_p$, with $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$. If $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}_a$ w.r.t. $\mathcal{Q}$, then $\bar{\mathcal{A}}_p$ is k-x-unmatchable in $\mathcal{A}_a$ w.r.t. $\mathcal{Q}$.*

Proof. By assumption there exists a re-identification σ_1 of $\mathcal{A}_p$ in $\mathcal{A}$ and a re-identification σ_2 of $\bar{\mathcal{A}}_p$ in $\mathcal{A}_p$. Consider $\sigma_3 = \sigma_1 \circ \sigma_2$. This is a substitution over $\bar{\mathcal{A}}_p$ and $\sigma_3(\bar{\mathcal{A}}_p) = \sigma_1(\sigma_2(\bar{\mathcal{A}}_p)) = \sigma_1(\mathcal{A}_p) = \mathcal{A}$. Furthermore, if $x = \mathsf{equi}$, clearly, $|\mathcal{A}| = |\mathcal{A}_p| = |\bar{\mathcal{A}}_p|$, while if $x = \mathsf{strict}$, then σ_1 and σ_2 are injective functions and so is σ_3. This proves that $\bar{\mathcal{A}}_p$ is an x-anonymization of $\mathcal{A}$. Since substitutions fix constant names, $\sigma_3(sig(\bar{\mathcal{A}}_p) \cap \mathcal{Q}) \subseteq sig(\mathcal{A}_p) \cap \mathcal{Q}$, but by definition of k-x-unmatchability, $sig(\mathcal{A}_p) \cap \mathcal{Q} = \emptyset$, so $sig(\bar{\mathcal{A}}_p) \cap \mathcal{Q} = \emptyset$. Now let $c \in \mathcal{Q}$, $b_1, \ldots, b_{k-1} \in sig(\bar{\mathcal{A}}_p) \cap \mathsf{N_B}$, and let $a_1 = \sigma_2(b_1), \ldots, a_{k-1} = \sigma_2(b_{k-1}) \in sig(\mathcal{A}_p)$. By assumption, there exists $\tau \in T_x[\mathcal{A}_p, \mathcal{A}_a]$ such that $c \notin \{\tau(\sigma_2(b_1), \ldots, \tau(\sigma_2(b_{k-1}))\}$. Consider $\bar{\tau} = \tau \circ \sigma_2$ and observe that $c \notin \{\bar{\tau}(b_1), \ldots, \bar{\tau}(b_{k-1})\}$. By definition of x-matching, there exists $\mathcal{A}'_p \subseteq \mathcal{A}_p$ that is an x-anonymization of $\mathcal{A}_a$ through τ. Then, consider $\bar{\mathcal{A}}_p{}' = \{\alpha \in \bar{\mathcal{A}}_p \mid \sigma_2(\alpha) \in \mathcal{A}'_p\}$, then clearly $\bar{\mathcal{A}}_p{}'$ is an x-anonymization of $\mathcal{A}'_p$ through σ_2. Then, by the same argument provided above, $\bar{\mathcal{A}}'_p$ is an x-anonymization of $\mathcal{A}_a$ through $\bar{\tau}$ and so $\bar{\tau} \in T_x[\bar{\mathcal{A}}_p, \mathcal{A}_a]$, hence the thesis. □

4 Data Complexity Analysis

We consider three decision problems, namely checking whether (i) a specific anonymization $\mathcal{A}_p$ of $\mathcal{A}$ is k-unmatchable w.r.t. the estimated knowledge of the attacker; (ii) such a k-unmatchable anonymization of $\mathcal{A}$ exists; (iii) a k-unmatchable anonymization of $\mathcal{A}$ exists whose cost is bounded by a threshold l. In this context, the parameter k represents a predetermined level of redundancy required of the data controller; we regard it as a constant and let only the ABoxes and the set of subjects $\mathcal{Q}$ vary (data complexity). Concerning the knowledge $\mathcal{A}_a$ possessed by the attacker, although in some scenarios $\mathcal{A}_a$ can be estimated, there is no universally applicable method to do it. Therefore, in our general theoretical analysis, we assume the worst-case scenario where $\mathcal{A}_a = \mathcal{A}$ (cf. Corollary 8). Thus, the above decision problems are formalized as follows.

Definition 10 (k-x-UNMATCHABILITY)**.** *Given an ABox $\mathcal{A}$, an x-anonymization $\mathcal{A}_p$ of $\mathcal{A}$ (with $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$) and a set $\mathcal{Q}$ of subjects in $\mathcal{A}$, decide whether $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$.*

Definition 11 (k-x-ANONYMIZATION)**.** *Given an ABox $\mathcal{A}$ and a set $\mathcal{Q}$ of subjects in $\mathcal{A}$, decide whether there exists an x-anonymization $\mathcal{A}_p$ of $\mathcal{A}$ (where $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$) such that $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$.*

In order to maximize the utility of the public ABox, the anonymization process should minimize the amount of concealed information $conc(\mathcal{A}_p)$, that we measure as in (Cuenca Grau and Kostylev 2019). Specifically, $conc(\mathcal{A}_p) = occ_{\mathsf{N_B}}(\mathcal{A}_p) + |sig(\mathcal{A}_p) \cap \mathsf{N_B}|$, where $occ_{\mathsf{N_B}}(\mathcal{A}_p)$ counts the number of occurrences of blanks in $\mathcal{A}_p$ (and $|sig(\mathcal{A}_p) \cap \mathsf{N_B}|$ is the number of distinct blanks used in $\mathcal{A}_p$).

Definition 12 (k-x-OPT-ANONYMIZATION)**.** *Given an ABox $\mathcal{A}$, a natural number l and a set $\mathcal{Q}$ of subjects in $\mathcal{A}$, decide whether there exists an x-anonymization $\mathcal{A}_p$ of $\mathcal{A}$ (where $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$) such that $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$ and $conc(\mathcal{A}_p) \leq l$.*

We now study the data complexity of these problems for $k \geq 2$ and $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$. Hereafter, when n is a natural number, we shall denote by $[n]$ the set $\{1, \ldots, n\}$. Complexity results are summarized in Table 1.

4.1 Upper bounds

In general k-x-UNMATCHABILITY is in NP, independently of the type of anonymization.

Theorem 13. *For all $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$, k-x-UNMATCHABILITY is in NP.*

Proof. K-unmatchability can be checked with the following nondeterministic algorithm: first check whether $sig(\mathcal{A}_p) \cap \mathcal{Q} = \emptyset$; then for all $c \in \mathcal{Q}$ and for all $\vec{b} = \langle b_1, \ldots, b_{k-1} \rangle \in (sig(\mathcal{A}_p) \cap \mathsf{N_B})^{k-1}$, guess a matching $\tau \in T_x[\mathcal{A}_p, \mathcal{A}]$ and check whether $c \notin \{\tau(b_1), \ldots, \tau(b_{k-1})\}$; accept the input iff all such tests succeed. Clearly, the above algorithm decides k-x-UNMATCHABILITY, i.e. it has an accepting run iff $(\mathcal{A}, \mathcal{A}_p, \mathcal{Q})$ is in k-x-UNMATCHABILITY. Moreover, the algorithm runs in polynomial time since (i) this is true of the first check, (ii) the number of all tuples $\langle c, \vec{b} \rangle$ is polynomial (bounded by $|\mathcal{Q}| \cdot |sig(\mathcal{A}_p)|^{k-1}$), (iii) each mapping τ consists of $|sig(\mathcal{A}_p)|$ pairs, and (iv) $c \notin \{\tau(b_1), \ldots, \tau(b_{k-1})\}$ can be checked in polynomial time. □

For strict-anonymizations we can refine this upper bound to GI ∩ NP. The following preliminary result is needed.

Lemma 14. *$\mathcal{A}_p$ is k-strict-unmatchable in $\mathcal{A}_a$ w.r.t. $\mathcal{Q}$ iff $sig(\mathcal{A}_p) \cap \mathcal{Q} = \emptyset$ and for each $c \in sig(\mathcal{A}_a) \cap \mathcal{Q}$ there exist at least k distinct blanks $b_1, \ldots, b_k \in sig(\mathcal{A}_p) \cap \mathsf{N_B}$ that are strict-matchable in c.*

	k-unmatchability	k-anonymization	k-opt-anonymization
strict anonymization	gi-hard/ GI ∩ NP	gi-hard/ GI ∩ NP	gi-hard/ NP
equipotent anonymization	NP-complete	? / LOGSPACE	gi-hard/ NP
poly anonymization	NP-complete	trivial	gi-hard/ NP

Table 1: Lower/upper data complexity bounds.

Proof. (Only if) Let $c \in \mathcal{Q} \cap sig(\mathcal{A}_a)$ and B_c be the set of blanks b in $\mathcal{A}_p$ that are strict-matchable in c (i.e. $\tau(b) = c$ for some $\tau \in T_{\mathsf{strict}}[\mathcal{A}_p, \mathcal{A}_a]$). Now, assume by contradiction that for some $c \in \mathcal{Q} \cap sig(\mathcal{A}_a)$, $|B_c| < k$. By definition of k-strict-unmatchability there exists $\tau \in T_{\mathsf{strict}}[\mathcal{A}_p, \mathcal{A}_a]$ such that $c \notin \{\tau(b) \mid b \in B_c\}$, a contradiction.

(If) Assume by contradiction that for some $c \in sig(\mathcal{A}_a) \cap \mathcal{Q}$, for some blanks $b_1, \ldots, b_{k-1}$ in $sig(\mathcal{A}_p) \cap \mathsf{N_B}$, and for all $\tau \in T_{\mathsf{strict}}[\mathcal{A}_p, \mathcal{A}_a]$, $c \in \{\tau(b_1), \ldots, \tau(b_{k-1})\}$. Then, let $b \in sig(\mathcal{A}_p) \cap \mathsf{N_B}$ be strict-matchable in c. This means that there exists $\tau \in T_{\mathsf{strict}}[\mathcal{A}_p, \mathcal{A}_a]$ such that $\tau(b) = c$. By assumption $\tau(b_i) = c$, for some $i = 1, \ldots, k-1$. Moreover, by injectivity, we also have that $b = b_i$. Then, the set of blanks of $\mathcal{A}_p$ that are strict-matchable in c has cardinality at most $k - 1$ (a contradiction). □

Theorem 15. k-strict-UNMATCHABILITY *is in GI.*

Proof (Sketch). Let $\mathcal{A}$ be an ABox, let $\mathcal{A}_p$ be a strict-anonymization of $\mathcal{A}$ and $\mathcal{Q}$ a set of constant names. For each $c \in \mathcal{Q}$ we define the ABox $\mathcal{A}^c = \mathcal{A} \cup \{F_c(c)\}$, where F_c is a fresh concept name. Moreover, for each blank b occurring in $\mathcal{A}_p$, let $\mathcal{A}_p^{c,b}$ be $\mathcal{A}_p \cup \{F_c(b)\}$. By Lemma 14, deciding whether $\mathcal{A}_p$ is k-strict-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$ reduces to verifying that $sig(\mathcal{A}_p) \cap \mathcal{Q} = \emptyset$ and that for each $c \in \mathcal{Q}$, the number of blanks occurring in $\mathcal{A}_p$ that are strict-matchable in c with $T_{\mathsf{strict}}[\mathcal{A}_p, \mathcal{A}]$ is at least k. It is straightforward to see that b is strict-matchable in c with $T_{\mathsf{strict}}[\mathcal{A}_p, \mathcal{A}]$ iff there exists a strict-matching of $\mathcal{A}_p^{c,b}$ in $\mathcal{A}^c$. Furthermore, finding a strict-matching between two ABoxes can be polynomial many-to-one reduced to graph isomorphism.[3] □

Remark 16. Unfortunately, the above proof cannot be readily extended to $x \in \{\mathsf{equi}, \mathsf{poly}\}$, because Lemma 14 is not valid for those x. To see this, let $\mathcal{A} = \{R(c, d_1), R(c, d_2)\}$, $\mathcal{A}_p = \{R(b_1, d_1), R(b_2, d_2)\}$, and $\mathcal{Q} = \{c\}$. The two blanks b_1, b_2 are matchable on c; however, since c is always mapped on both blanks by all matchings, $\mathcal{A}_p$ (which is both an equi-anonymization and a poly-anonymization of $\mathcal{A}$) is not 2-x-unmatchable in $\mathcal{A}$, for any $x \in \{\mathsf{equi}, \mathsf{poly}\}$. □

Next, we analyze k-x-ANONYMIZATION. For $x = \mathsf{strict}$, it is in GI ∩ NP like k-strict-UNMATCHABILITY because:

Theorem 17. k-strict-ANONYMIZATION *is polynomial many-to-one reducible to* k-strict-UNMATCHABILITY.

Proof. For each ABox $\mathcal{A}$, define ABox $\mathcal{A}^g$ by uniformly substituting each individual $a \in sig(\mathcal{A})$ with a fresh blank b^a. Note that the substitution τ_g such that $\tau_g(b) = a$ if $b = b^a$ (for all blanks $b \in sig(\mathcal{A}^g)$) is injective and satisfies $\tau_g(\mathcal{A}^b) = \mathcal{A}$, i.e. $\mathcal{A}^g$ is a strict-anonymization of $\mathcal{A}$.

Let $\mathcal{A}'$ be any strict-anonymization of $\mathcal{A}$ through some injective τ'. Define a substitution $\hat{\tau}$ from $\mathcal{A}^g$ to $\mathcal{A}'$ so that, for each blank b^a occurring in $\mathcal{A}^g$, $\hat{\tau}(b^a) = b$ if $\tau'(b) = a$; $\hat{\tau}(b^a) = a$, otherwise. It is straightforward to see that $\mathcal{A}^g$ is a strict-anonymization of $\mathcal{A}'$ through $\hat{\tau}$. Then, by Theorem 9, $\mathcal{A}$ has a k-strict-anonymization $\mathcal{A}'$ w.r.t. $\mathcal{Q}$ iff $\mathcal{A}^g$ is k-strict-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$. Clearly, $\mathcal{A}^g$ can be computed in polynomial time. □

For $x = \mathsf{equi}$ we have a better (LOGSPACE) upper bound, whose proof needs auxiliary definitions. From each ABox $\mathcal{A}$ obtain an ABox $\mathcal{A}^*$ by replacing each *occurrence* of each individual a in $sig(\mathcal{A})$ with a distinct fresh blank b_i^a – where index i ranges from 1 to the number of occurrences of a in $\mathcal{A}$. By construction, $\mathcal{A}^*$ is an equi-anonymization of $\mathcal{A}$ through the substitution $\tau_{\mathcal{A}}$ such that $\tau_{\mathcal{A}}(b) = a$ if $b = b_i^a$. Our first result says that in order to solve k-equi-ANONYMIZATION for $\mathcal{A}$ it suffices to test the unmatchability of $\mathcal{A}^*$ in $\mathcal{A}$.

Lemma 18. *If $\mathcal{A}_p$ is k-equi-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$, then also $\mathcal{A}^*$ is k-equi-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$.*

Proof. For all individuals $a \in sig(\mathcal{A})$, and each occurrence of a in $\mathcal{A}$, if that occurrence is translated to b_i^a in $\mathcal{A}^*$, then let a_i^p denote the translation of that same occurrence in $\mathcal{A}_p$. For all blanks $b_i^a \in sig(\mathcal{A}^*)$, let $\tau_0(b_i^a) = a_i^p$. By construction, $\tau_0(\mathcal{A}^*) = \mathcal{A}_p$, so $\mathcal{A}^*$ is an equi-anonymization of $\mathcal{A}_p$. Then the Lemma follows from Theorem 9. □

We need also a second lemma and auxiliary notation: for all predicates $P \in \mathsf{N_C} \cup \mathsf{N_R}$ and all ABoxes $\mathcal{A}$, $\#(P, \mathcal{A})$ denotes the number of occurrences of P in $\mathcal{A}$; for all $c \in \mathcal{Q}$ and $i \in \{1, 2\}$, $\#(P, c, i, \mathcal{A})$ denotes the number of atoms with predicate P where c occurs as i-th argument. Note that if P is unary, then $\#(P, c, 1, \mathcal{A}) \leq 1$.

Lemma 19. *$\mathcal{A}^*$ is k-equi-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$ iff, for all $P \in \mathsf{N_C} \cup \mathsf{N_R}$, $c \in \mathcal{Q}$, and $i \in \{1, 2\}$ such that $\#(P, c, i, \mathcal{A}) > 0$, $\#(P, \mathcal{A}) \geq k - 1 + \#(P, c, i, \mathcal{A})$.*

Proof. (If) Assume that $\#(P, c, i, \mathcal{A}) > 0$ implies $\#(P, \mathcal{A}) \geq k - 1 + \#(P, c, i, \mathcal{A})$. Let c be any subject in $\mathcal{Q}$ and $b_1, \ldots, b_{k-1}$ be any blanks in $sig(\mathcal{A}^*)$. We have to prove that there exists an equi-matching τ of $\mathcal{A}^*$ in $\mathcal{A}$ such that $c \notin \{\tau(b_1), \ldots, \tau(b_{k-1})\}$. Such τ can be constructed incrementally as follows, starting with unary axioms.

By assumption, for each unary axiom $P(c) \in \mathcal{A}$ there exist at least $k - 1 + \#(P, c, 1, \mathcal{A}) = k$ axioms of the form $P(a) \in \mathcal{A}$. For each of these axioms, $\mathcal{A}^*$ contains a distinct axiom $P(b)$. So $\mathcal{A}^*$ must contain at least one axiom $P(b')$

[3] The reduction will be described in an extended version of this paper.

where $b' \notin \{b_1, \ldots, b_{k-1}\}$. Choose one of such axioms and set $\tau(b') = c$.

Next τ is extended to the arguments of binary axioms in a similar way: by assumption, for each axiom of the form $P(c, a)$ there exist at least $n = k - 1 + \#(P, c, 1, \mathcal{A})$ axioms of the form $P(a'_i, a''_i) \in \mathcal{A}$ $(1 \leq i \leq \#(P, \mathcal{A}))$. So $\mathcal{A}^*$ must contain at least $\#(P, c, 1, \mathcal{A})$ axioms of the form $P(b'_i, b''_i)$ where $b'_i \notin \{b_1, \ldots, b_{k-1}\}$. Choose a set S containing $\#(P, c, 1, \mathcal{A})$ of such axioms and set τ so that $\tau(S)$ is the set of all axioms of the form $P(c, a')$ in $\mathcal{A}$ (this is always possible because every blank in $sig(\mathcal{A}^*)$ occurs in $\mathcal{A}^*$ exactly once). Deal with axioms of the form $P(a, c)$ in a similar way, each time picking axioms from $\mathcal{A}^*$ whose arguments are not yet in the partial domain of τ.

The partial definition of τ specified above maps a subset of $\mathcal{A}^*$ onto all the axioms of $\mathcal{A}$ involving c, without assigning any values to $b_1, \ldots, b_{k-1}$. Now any predicate-preserving extension of this partial definition of τ that maps each of the remaining axioms of $\mathcal{A}^*$ on any of the axioms of $\mathcal{A}$ *not* involving c is clearly an equi-matching of $\mathcal{A}^*$ in $\mathcal{A}$. Moreover, by construction, $c \notin \{\tau(b_1), \ldots, \tau(b_{k-1})\}$.

(Only if) Assume that for some P, c, i of the appropriate type, $\#(P, c, i, \mathcal{A}) > 0$ but $\#(P, \mathcal{A}) < k-1+\#(P, c, i, \mathcal{A})$.

Let $B_i^P = \{b_1, \ldots, b_{\#(P,\mathcal{A})}\}$ be the blanks occurring as the i-th arguments of the axioms of $\mathcal{A}^*$ with predicate P. Let $n = \#(P, \mathcal{A}) - \#(P, c, i, \mathcal{A})$ and note that $n < k - 1$.

Now let τ be any equi-matching of $\mathcal{A}^*$ in $\mathcal{A}$. τ must map $\#(P, c, i, \mathcal{A})$ axioms with predicate P onto the axioms of $\mathcal{A}$ with predicate P where c occurs in the i-th position. Then τ shall map $\#(P, c, i, \mathcal{A})$ of the blanks in B_i^P on c. Since the remaining blanks in B_i^P are $n < k - 1$, it follows that for some $j \in [1, m]$, where $m = \min(k - 1, \#(P, \mathcal{A}))$, $\tau(b_j) = c$. Since τ is an arbitrary equi-matching, we conclude that for all such matchings, $c \in \{\tau(b_1), \ldots, \tau(b_m)\}$, where $m \leq k - 1$. It follows immediately that $\mathcal{A}^*$ is not k-equi-unmatchable. □

Theorem 20. k-equi-ANONYMIZATION *is in LOGSPACE.*

Proof. By Lemmas 18 and 19, $\mathcal{A}$ has a k-equi-unmatchable equi-anonymization iff, for all $P \in \mathsf{N_C} \cup \mathsf{N_R}$, $c \in \mathcal{Q}$, and $i \in \{1, 2\}$ such that $\#(P, c, i, \mathcal{A}) > 0$, $\#(P, \mathcal{A}) \geq k - 1 + \#(P, c, i, \mathcal{A})$.

Note that each of P, c, i, $\#(P, \mathcal{A})$, and $\#(P, c, i, \mathcal{A})$ may take at most n values, where n is the size of $\mathcal{A}$, so they can be encoded with $\log(n)$ bits, and the above test can be computed in space $O(\log n)$. □

We are left to analyze k-poly-ANONYMIZATION, which turns out to be trivial.

Theorem 21. *Let $\mathcal{A}$ be an ABox and $\mathcal{Q}$ a set of subjects in $\mathcal{A}$, then $\mathcal{A}$ has a k-poly-unmatchable anonymization $\mathcal{A}^*_{\mathsf{poly}}$ w.r.t. $\mathcal{Q}$, such that $|\mathcal{A}^*_{\mathsf{poly}}| = k \cdot |\mathcal{A}|$ and $occ_{\mathsf{N_B}}(\mathcal{A}^*_{\mathsf{poly}}) = |sig(\mathcal{A}^*_{\mathsf{poly}}) \cap \mathsf{N_B}| \leq 2k \cdot |\mathcal{A}|$.*

Proof. Let $\mathcal{A}^*_{\mathsf{poly}} = \mathcal{A}^*_1 \cup \ldots \cup \mathcal{A}^*_k$, where each $\mathcal{A}^*_i$ is a renaming of the $\mathcal{A}^*$ used in the proofs of Lemma 18, where $sig(A^*_i) \cap sig(A^*_j) = \emptyset$ $(1 \leq i < j \leq k)$. Recall that by construction each $\mathcal{A}^*_i$ is a poly-anonymization of $\mathcal{A}$.

By construction, $|\mathcal{A}^*_{\mathsf{poly}}| = k \cdot |\mathcal{A}|$. To prove that $\mathcal{A}^*_{\mathsf{poly}}$ is a k-poly-anonymization of $\mathcal{A}$ first note that $\mathcal{A}^*_{\mathsf{poly}}$ satisfies point (i) of Definition 5 by construction, because it contains only blanks. Point (ii) of Definition 5 can easily be proved by leveraging the k copies of $\mathcal{A}^*$ included in $\mathcal{A}^*_{\mathsf{poly}}$ (the details are left to the reader due to space limitations). Thus $\mathcal{A}^*_{\mathsf{poly}}$ is a k-poly-anonymization of $\mathcal{A}$ w.r.t. $\mathcal{Q}$.

Finally, note that: (i) every blank occurs at most once in $\mathcal{A}^*_{\mathsf{poly}}$, so $occ_{\mathsf{N_B}}(\mathcal{A}^*_{\mathsf{poly}}) = |sig(\mathcal{A}^*_{\mathsf{poly}}) \cap \mathsf{N_B}|$; (ii) each $\mathcal{A}^*_i$ contains at most $2 \cdot |\mathcal{A}|$ distinct blanks (the limit is reached when all axioms in $\mathcal{A}$ are binary), hence the bound $|sig(\mathcal{A}^*_{\mathsf{poly}}) \cap \mathsf{N_B}| \leq 2k \cdot |\mathcal{A}|$. □

Finally, consider k-x-OPT-ANONYMIZATION. Theorem 21 implies a bound on the cardinality of poly-anonymizations whose cost is bound by a constant ℓ.

Lemma 22. *For all integers ℓ, if $\mathcal{A}$ has a* poly*-anonymization $\mathcal{A}_p$ which is k-poly-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$, such that $conc(\mathcal{A}_p) \leq \ell$, then there exists $\mathcal{A}'_p$ with the same properties as $\mathcal{A}_p$ and such that $|\mathcal{A}'_p| \leq 2k^{k-1}|\mathcal{Q}| \cdot |\mathcal{A}|^k$.*

Proof. By Theorem 21, every ABox $\mathcal{A}$ has a k-poly-unmatchable anonymization $\mathcal{A}^*_{\mathsf{poly}}$ such that $conc(\mathcal{A}^*_{\mathsf{poly}}) \leq 4k \cdot |\mathcal{A}|$. Thus, if ℓ is greater than, or equal to this bound, then the Lemma is proved with $\mathcal{A}'_p = \mathcal{A}^*_{\mathsf{poly}}$. Now assume that $\ell < 4k \cdot |\mathcal{A}|$ and note that $conc(\mathcal{A}_p) < 4k \cdot |\mathcal{A}|$ implies that $|sig(\mathcal{A}_p) \cap \mathsf{N_B}| \leq conc(\mathcal{A}_p)/2 < 2k \cdot |\mathcal{A}|$ (because $|sig(\mathcal{A}_p) \cap \mathsf{N_B}| \leq occ_{\mathsf{N_B}}(\mathcal{A}_p)$).

Take any poly-anonymization $\mathcal{A}_p$ of $\mathcal{A}$ that is k-poly-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$ and such that $conc(\mathcal{A}_p) \leq \ell$. By k-unmatchability, $\mathcal{A}_p$ contains no subjects from $\mathcal{Q}$ and, for all $c \in \mathcal{Q}$ and $B = \{b_1, \ldots, b_{k-1}\} \subseteq sig(\mathcal{A}_p) \cap \mathsf{N_B}$, there is a poly-matching $\tau_{c,B}$ of $\mathcal{A}_p$ in $\mathcal{A}$ such that $c \notin \{\tau_{c,B}(b_1), \ldots, \tau_{c,B}(b_{k-1})\}$. For each $\tau_{c,B}$, let $\mathcal{A}_{c,B}$ denote the subset of $\mathcal{A}_p$ such that $\tau_{c,B}(\mathcal{A}_{c,B}) = \mathcal{A}$ (which exists by definition of matching).

Let $\mathcal{A}'_p = \bigcup_{c,B} \mathcal{A}_{c,B}$. Note that $\mathcal{A}'_p$ is k-poly-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$, since it is a subset of $\mathcal{A}_p$ (which contains no subjects from $\mathcal{Q}$), and for all $c \in \mathcal{Q}$ and $B = \{b_1, \ldots, b_{k-1}\} \subseteq sig(\mathcal{A}'_p) \cap \mathsf{N_B}$, there is a matching $\tau'_{c,B}$ of $\mathcal{A}'_p$ in $\mathcal{A}$ such that $c \notin \{\tau'_{c,B}(b_1), \ldots, \tau'_{c,B}(b_{k-1})\}$ (each $\tau'_{c,B}$ is simply the restriction of $\tau_{c,B}$ to $\mathcal{A}'_p$).

We are only left to assess the size of $\mathcal{A}'_p$. Each $\mathcal{A}_{c,B}$ can be assumed to have the same cardinality as $\mathcal{A}$, without loss of generality (by removing redundant axioms). Thus $|\mathcal{A}'_p| \leq |\mathcal{Q}| \cdot |sig(\mathcal{A}_p) \cap \mathsf{N_B}|^{k-1} \cdot |\mathcal{A}| \leq |\mathcal{Q}| \cdot (2k \cdot |\mathcal{A}|)^{k-1} \cdot |\mathcal{A}| \leq 2k^{k-1} \cdot |\mathcal{Q}| \cdot |\mathcal{A}|^k$. □

This yields an upper bound for all optimal anonymizations.

Theorem 23. *For each $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$, k-x-*OPT-ANONYMIZATION *is in NP.*

Proof. The following is a nondeterministic polynomial algorithm that decides k-x-OPT-ANONYMIZATION. First guess an x-anonymization $\mathcal{A}_p$ of the given ABox $\mathcal{A}$. Then apply the polynomial nondeterministic algorithm illustrated in the proof of Theorem 13 to check whether $\mathcal{A}_p$ is k-x-unmatchable in $\mathcal{A}$ w.r.t. the given $\mathcal{Q}$. Finally, compute

$conc(\mathcal{A}_p)$ and check whether it is bound by the given threshold l.

The first guess takes polynomial time, since for x in $\{\mathsf{strict}, \mathsf{equi}\}$, $|\mathcal{A}_p| = |\mathcal{A}|$, and for $x = \mathsf{poly}$, we can assume a polynomial bound on cardinality by Lemma 22. We have already proved that k-unmatchability can be checked in polynomial nondeterministic time. Clearly, checking whether $conc(\mathcal{A}_p) \leq l$ can be done in polynomial time. □

4.2 Lower bounds

Due to space limitations, we provide detailed proofs for $k = 2$. The proofs for the general case $k \geq 2$ use straightforward but lengthy extensions of the encodings, so they will be described in an extended version of this paper.

We start by proving that *graph isomorphism* can be reduced to k-strict-UNMATCHABILITY and *subgraph isomorphism* to k-x-UNMATCHABILITY for $x \in \{\mathsf{poly}, \mathsf{equi}\}$. For this purpose, we encode any given pair of graphs G and H with the ABox $\mathcal{A}(G, H)$ illustrated in Table 2. Constants $\mathsf{g_G}$ and $\mathsf{g_H}$ represent graphs G and H, respectively, and are marked by the concept "graph". Each vertex $v \in V_G$ (resp. $w \in V_H$) is represented by constant $\mathsf{c_v}$ (resp. $\mathsf{c_w}$).[4] Graph edges are encoded by role "edge", whereas role "node" associates vertices to the graph they belong to.

For each $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$, we shall define an x-anonymization of $\mathcal{A}(G, H)$ that is k-x-unmatchable iff G and H are in a suitable isomorphism relation.

We start with $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ (defined in Table 2), a strict-anonymization of $\mathcal{A}(G, H)$ that replaces blanks $\mathsf{bg_G}$ and $\mathsf{bg_H}$ for $\mathsf{g_G}$ and $\mathsf{g_H}$, respectively, while blanks $\mathsf{b_v}$ and $\mathsf{b_w}$ replace $\mathsf{c_v}$ and $\mathsf{c_w}$, respectively. $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ is a strict-anonymization of $\mathcal{A}(G, H)$ through the following injective re-identification.

Definition 24. *Let G, H be graphs, we define the substitution $\tau_a^{G,H}$ over $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ as follows:*

$$\begin{array}{ll} \tau_a^{G,H}(\mathsf{b_v}) = \mathsf{c_v} & \forall v \in V_G \\ \tau_a^{G,H}(\mathsf{b_w}) = \mathsf{c_w} & \forall w \in V_H \\ \tau_a^{G,H}(\mathsf{bg_G}) = \mathsf{g_G} & \\ \tau_a^{G,H}(\mathsf{bg_H}) = \mathsf{g_H} & \end{array}$$

The next two lemmas show that the isomorphisms of G in H correspond to the class of strict-matchings of $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ in $\mathcal{A}(G, H)$ that map $\mathsf{bg_H}$ in $\mathsf{g_G}$.

Lemma 25. *Let G, H be graphs and let φ be a graph isomorphism from G to H, then there exists a* strict*-matching τ_φ of $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ in $\mathcal{A}(G, H)$ such that $\tau_\varphi(\mathsf{bg_H}) = \mathsf{g_G}$.*

Proof (Sketch). Let substitution τ_φ on $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ be:

$$\begin{array}{ll} \tau_\varphi(\mathsf{b_v}) = \mathsf{c}_{\varphi(\mathsf{v})} & \forall v \in V_G \\ \tau_\varphi(\mathsf{b_w}) = \mathsf{c}_{\varphi^{-1}(\mathsf{w})} & \forall w \in V_H \\ \tau_\varphi(\mathsf{bg_G}) = \mathsf{g_H} & \\ \tau_\varphi(\mathsf{bg_H}) = \mathsf{g_G} & \end{array}$$

This substitution uses graph isomorphism φ to map the blanks corresponding to the vertices of G in the constant names corresponding to the vertices of H and the blanks corresponding to the vertices of H in the constant names corresponding to the vertices of G. This ensures that τ_φ is an injective re-identification of $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ in $\mathcal{A}(G, H)$. □

[4] We assume w.l.o.g. that G and H have disjoint sets of vertices.

Lemma 26. *Let G, H be graphs and let τ be a* strict*-matching of $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ in $\mathcal{A}(G, H)$ such that $\tau(\mathsf{bg_H}) \neq \mathsf{g_H}$, then G and H are isomorphic.*

Proof (Sketch). Note that $\mathcal{A}(G, H)$ only contains two atoms using the concept graph: $\mathsf{graph}(\mathsf{g_G})$ and $\mathsf{graph}(\mathsf{g_H})$. So, since τ is a strict-matching of $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ in $\mathcal{A}(G, H)$ and $\tau(\mathsf{bg_H}) \neq \mathsf{g_H}$, the atom $\mathsf{graph}(\mathsf{bg_H})$ must be mapped by τ in the atom $\mathsf{graph}(\mathsf{g_G})$, therefore $\tau(\mathsf{bg_H}) = \mathsf{g_G}$. This implies that all atoms of type $\mathsf{node}(\mathsf{bg_H}, a)$ must by mapped by τ in atoms of type $\mathsf{node}(\mathsf{g_G}, a')$. The injectivity of τ and the fact that it is a matching thus establishes a bijection between vertices of G and the vertices of H. Morover this bijection is a graph isomorphism, because atoms of type edge are mapped in atoms of type edge by τ, hence adjacent vertices are mapped in adjacent vertices and vice versa. □

From these lemmata it follows that:

Theorem 27. k-strict-UNMATCHABILITY *is gi-hard.*

Proof (Sketch). Let G, H be graphs, we build the ABoxes $\mathcal{A}(G, H)$ and $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ in polynomial time. Let $\mathcal{Q} = sig(\mathcal{A}(G, H))$. Now, assume there exists a graph isomorphism φ of G in H and let us prove that $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ is 2-strict-unmatchable in $\mathcal{A}(G, H)$ w.r.t. $\mathcal{Q}$. Assume by contradiction there exist $c \in \mathcal{Q}$ and $b \in sig(\mathcal{A}_p(G, H)^{\mathsf{strict}}) \cap \mathsf{N_B}$ such that for each $\tau \in T_{\mathsf{strict}}[\mathcal{A}_p(G, H)^{\mathsf{strict}}, \mathcal{A}(G, H)]$, $\tau(b) = c$. Now, consider the strict-matching $\tau_a^{G,H}$ of Definition 24 and the strict-matching τ_φ obtained as in Lemma 25. Observe that for each $b' \in sig(\mathcal{A}_p(G, H)^{\mathsf{strict}}) \cap \mathsf{N_B}$, $\tau_a^{G,H}(b') \neq \tau_\varphi(b')$, but by assumption $\tau_a^{G,H}(b) = \tau_\varphi(b) = c$, hence an absurd. Conversely, assume that $\mathcal{A}_p(G, H)^{\mathsf{strict}}$ is 2-strict-unmatchable in $\mathcal{A}(G, H)$ w.r.t. $\mathcal{Q}$ and note that $\mathsf{bg_H} \in sig(\mathcal{A}_p(G, H)^{\mathsf{strict}})$. So, by definition of k-strict-unmatchability, there is $\tau \in T_{\mathsf{strict}}[\mathcal{A}_p(G, H)^{\mathsf{strict}}, \mathcal{A}(G, H)]$ such that $\tau(\mathsf{bg_H}) \neq \mathsf{g_H}$, hence by Lemma 26, there exists a graph isomorphism from G to H. This completes the proof for $k = 2$. The proof for $k > 2$ can be obtained by encoding $k - 1$ copies of G in $\mathcal{A}(G, H)$ and $\mathcal{A}_p(G, H)^{\mathsf{strict}}$, and adapting the above proofs accordingly. □

For $x \in \{\mathsf{equi}, \mathsf{poly}\}$, we define the x-anonymization $\mathcal{A}_p(G, H)^{\mathsf{equi}}$ of $\mathcal{A}(G, H)$ illustrated in Table 2, which uses a pair of blanks $\mathsf{b_e^1}, \mathsf{b_e^2}$ for each edge $e \in E_G$, a blank $\mathsf{bg_G^v}$ and a blank $\mathsf{b_v}$ for each $v \in V_G$, a blank $\mathsf{b_w}$ for each $w \in V_H$ and the blanks $\mathsf{bg_H}, \mathsf{bg_G}$.

This is an equi-anonymization and a poly-anonymization through the following following substitution.

ABox $\mathcal{A}(G,H)$		ABox $\mathcal{A}_p(G,H)^{\mathsf{strict}}$		ABox $\mathcal{A}_p(G,H)^{\mathsf{equi}}$	
$\mathsf{edge}(\mathsf{c}_{\mathsf{v}_1}, \mathsf{c}_{\mathsf{v}_2})$,	$\forall (v_1, v_2) \in E_G$	$\mathsf{edge}(\mathsf{b}_{\mathsf{v}_1}, \mathsf{b}_{\mathsf{v}_2})$,	$\forall (v_1, v_2) \in E_G$	$\mathsf{edge}(\mathsf{b}^1_{\mathsf{e}}, \mathsf{b}^2_{\mathsf{e}})$,	$\forall e \in E_G$
$\mathsf{edge}(\mathsf{c}_{\mathsf{w}_1}, \mathsf{c}_{\mathsf{w}_2})$,	$\forall (w_1, w_2) \in E_H$	$\mathsf{edge}(\mathsf{b}_{\mathsf{w}_1}, \mathsf{b}_{\mathsf{w}_2})$,	$\forall (w_1, w_2) \in E_H$	$\mathsf{edge}(\mathsf{b}_{\mathsf{w}_1}, \mathsf{b}_{\mathsf{w}_2})$,	$\forall (w_1, w_2) \in E_H$
$\mathsf{node}(\mathsf{g}_{\mathsf{G}}, \mathsf{c}_{\mathsf{v}})$,	$\forall v \in V_G$	$\mathsf{node}(\mathsf{bg}_{\mathsf{G}}, \mathsf{b}_{\mathsf{v}})$,	$\forall v \in V_G$	$\mathsf{node}(\mathsf{bg}^{\mathsf{v}}_{\mathsf{G}}, \mathsf{b}_{\mathsf{v}})$,	$\forall v \in V_G$
$\mathsf{node}(\mathsf{g}_{\mathsf{H}}, \mathsf{c}_{\mathsf{w}})$,	$\forall w \in V_H$	$\mathsf{node}(\mathsf{bg}_{\mathsf{H}}, \mathsf{b}_{\mathsf{w}})$,	$\forall w \in V_H$	$\mathsf{node}(\mathsf{bg}_{\mathsf{H}}, \mathsf{b}_{\mathsf{w}})$,	$\forall w \in V_H$
$\mathsf{graph}(\mathsf{g}_{\mathsf{G}})$		$\mathsf{graph}(\mathsf{bg}_{\mathsf{G}})$		$\mathsf{graph}(\mathsf{bg}_{\mathsf{G}})$	
$\mathsf{graph}(\mathsf{g}_{\mathsf{H}})$.		$\mathsf{graph}(\mathsf{bg}_{\mathsf{H}})$.		$\mathsf{graph}(\mathsf{bg}_{\mathsf{H}})$.	

Table 2: ABoxes used in Section 4.2, parameterised by graphs G and H

Definition 28. *For all graph pairs G, H let $\tau_a^{G,H}$ be defined as:*

$$\begin{array}{ll}
\tau_a^{G,H}(\mathsf{b}_{\mathsf{v}}) = \mathsf{c}_{\mathsf{v}} & \forall v \in V_G \\
\tau_a^{G,H}(\mathsf{b}_{\mathsf{w}}) = \mathsf{c}_{\mathsf{w}} & \forall w \in V_H \\
\tau_a^{G,H}(\mathsf{bg}_{\mathsf{G}}) = \mathsf{g}_{\mathsf{G}} & \\
\tau_a^{G,H}(\mathsf{bg}_{\mathsf{H}}) = \mathsf{g}_{\mathsf{H}} & \\
\tau_a^{G,H}(\mathsf{b}^1_{\mathsf{e}}) = \mathsf{c}_{\mathsf{v}} & \forall e = (v, v') \in E_G \\
\tau_a^{G,H}(\mathsf{b}^2_{\mathsf{e}}) = \mathsf{c}_{\mathsf{v}'} & \forall e = (v, v') \in E_G \\
\tau_a^{G,H}(\mathsf{bg}^{\mathsf{v}}_{\mathsf{G}}) = \mathsf{g}_{\mathsf{G}} & \forall v \in V_G
\end{array}$$

Analogously to the strict case, there is a correspondence between the subgraph isomorphisms of H in G and the x-matchings (with $x \in \{\mathsf{equi}, \mathsf{poly}\}$) of $\mathcal{A}_p(G,H)^{\mathsf{equi}}$ in $\mathcal{A}(G,H)$ that do not map bg_{H} in g_{H}. First we recall some definitions and properties about subgraph isomorphism.

Definition 29. *Let G and H be graphs; a subgraph isomorphism of H in G is a pair (G', f) where G' is a subgraph of G, i.e. $V_{G'} \subseteq V_G$ and $E_{G'} \subseteq E_G \cap (V_{G'} \times V_{G'})$, and f is a graph isomorphism between G' and H.*

The following proposition establishes a well known sufficient condition for the existence of a subgraph isomorphism.

Proposition 30. *Let G and H be graphs and let $h : V_H \to V_G$ be an injective function such that for all $(v, w) \in E_H$, $(h(v), h(w)) \in E_G$. Then, there exists a subgraph isomorphism f of H in G.*

Then, we can prove the following preliminary results.

Lemma 31. *Let G, H be graphs and let φ be a subgraph isomorphism of H in G. For all $x \in \{\mathsf{equi}, \mathsf{poly}\}$, there exists an x-matching τ_φ of $\mathcal{A}_p(G,H)^{\mathsf{equi}}$ in $\mathcal{A}(G,H)$ such that $\tau_\varphi(\mathsf{bg}_{\mathsf{H}}) = \mathsf{g}_{\mathsf{G}}$.*

Proof (Sketch). Let $\varphi = (G', f)$, where G' is a subgraph of G and $f : G' \to H$ is a graph isomorphism. We define τ_φ as follows.

$$\begin{array}{ll}
\tau_\varphi(\mathsf{b}_{\mathsf{v}}) = \mathsf{c}_{\mathsf{f(v)}} & \forall v \in V_{G'} \\
\tau_\varphi(\mathsf{b}_{\mathsf{v}}) = \mathsf{c}_{\mathsf{v}} & \forall v \in V_G \setminus V_{G'} \\
\tau_\varphi(\mathsf{b}_{\mathsf{w}}) = \mathsf{c}_{\mathsf{f^{-1}(w)}} & \forall w \in V_H \\
\tau_\varphi(\mathsf{bg}_{\mathsf{H}}) = \mathsf{g}_{\mathsf{G}} & \\
\tau_\varphi(\mathsf{bg}^{\mathsf{v}}_{\mathsf{G}}) = \mathsf{g}_{\mathsf{H}} & \forall v \in V_{G'} \\
\tau_\varphi(\mathsf{bg}^{\mathsf{v}}_{\mathsf{G}}) = \mathsf{g}_{\mathsf{G}} & \forall v \in V_G \setminus V_{G'} \\
\tau_\varphi(\mathsf{b}^1_{\mathsf{e}}) = \mathsf{c}_{\mathsf{v}} & \forall e = (v, v') \in E_G \setminus E_{G'} \\
\tau_\varphi(\mathsf{b}^1_{\mathsf{e}}) = \mathsf{c}_{\mathsf{f(v)}} & \forall e = (v, v') \in E_{G'} \\
\tau_\varphi(\mathsf{b}^2_{\mathsf{e}}) = \mathsf{c}_{\mathsf{v}'} & \forall e = (v, v') \in E_G \setminus E_{G'} \\
\tau_\varphi(\mathsf{b}^2_{\mathsf{e}}) = \mathsf{c}_{\mathsf{f(v')}} & \forall e = (v, v') \in E_{G'} \\
\tau_\varphi(\mathsf{bg}_{\mathsf{G}}) = \mathsf{g}_{\mathsf{H}} &
\end{array}$$

One can easily check that τ_φ is an equi-matching (and so a poly-matching too). The main observation is that atoms regarding vertices of H are mapped in atoms regarding vertices of G' and vice versa; while for other atoms, τ_φ behaves as $\tau_a^{G,H}$. □

Lemma 32. *Let $\mathcal{A}_1, \mathcal{A}_2$ be two ABoxes such that $|\mathcal{A}_1| = |\mathcal{A}_2|$ and let τ be an x-matching of $\mathcal{A}_1$ in $\mathcal{A}_2$, with $x \in \{\mathsf{equi}, \mathsf{poly}\}$. If $\alpha_1, \alpha_2 \in \mathcal{A}_1$ are such that $\tau(\alpha_1) = \tau(\alpha_2)$, then $\alpha_1 = \alpha_2$.*

Proof (Sketch). Let $x \in \{\mathsf{equi}, \mathsf{poly}\}$, by definition of x-matching, there exists $\mathcal{A}'_1 \subseteq \mathcal{A}_1$ such that $\mathcal{A}'_1$ is an x-anonymization of $\mathcal{A}_2$ through τ and so $\tau(\mathcal{A}'_1) = \mathcal{A}_2$. By the fact that $|\mathcal{A}_2| = |\tau(\mathcal{A}'_1)| \leq |\mathcal{A}'_1| \leq |\mathcal{A}_1|$ and by the assumption that $|\mathcal{A}_1| = |\mathcal{A}_2|$, we can deduce that $\mathcal{A}'_1 = \mathcal{A}_1$. Then, since $|\mathcal{A}_1| = |\mathcal{A}_2| = |\tau(\mathcal{A}_1)|$, we get the thesis. □

Lemma 33. *Let G, H be graphs and $x \in \{\mathsf{equi}, \mathsf{poly}\}$. Assume there exists an x-matching τ of $\mathcal{A}_p(G,H)^{\mathsf{equi}}$ in $\mathcal{A}(G,H)$ such that $\tau(\mathsf{bg}_{\mathsf{H}}) \neq \mathsf{g}_{\mathsf{H}}$, then there exists a subgraph isomorphism of H in G.*

Proof. By assumption τ is an x-matching of $\mathcal{A}_p(G,H)^{\mathsf{equi}}$ in $\mathcal{A}(G,H)$, so $\tau(\mathsf{graph}(\mathsf{bg}_{\mathsf{H}})) \in \mathcal{A}(G,H)$, this means that $\mathcal{A}(G,H)$ contains an atom $\alpha = \mathsf{graph}(a)$ for some $a \in \mathsf{N_I}$. By construction, a can either be g_{G} or g_{H}. But by assumption $\tau(\mathsf{bg}_{\mathsf{H}}) \neq \mathsf{g}_{\mathsf{H}}$, therefore it must be that $\tau(\mathsf{bg}_{\mathsf{H}}) = \mathsf{g}_{\mathsf{G}}$. Now, let $w \in V_H$ and consider the atom $\beta = \mathsf{graph}(\mathsf{bg}_{\mathsf{H}}, \mathsf{b}_{\mathsf{w}})$. Since τ is an x-matching of $\mathcal{A}_p(G,H)^{\mathsf{equi}}$ in $\mathcal{A}(G,H)$, there exists an atom $\beta' \in \mathcal{A}(G,H)$ such that $\tau(\beta) = \beta'$, therefore β' must be of the form $\mathsf{node}(\mathsf{g}_{\mathsf{G}}, a)$, where $a \in \mathsf{N_I}$. Furthermore, by definition of $\mathcal{A}(G,H)$, a is of type c_{v}, for some $v \in V_G$. Therefore for each $w \in V_H$, there exists a unique $v \in V_G$ such that $\tau(\mathsf{b}_{\mathsf{w}}) = \mathsf{c}_{\mathsf{v}}$. We can thus define a function φ mapping each $w \in V_H$ to the unique $v \in V_G$ such that $\tau(\mathsf{b}_{\mathsf{w}}) = \mathsf{c}_{\mathsf{v}}$. Observe that φ is injective, in fact assume that there exist $w_1, w_2 \in V_H$ such that $w_1 \neq w_2$ but $\varphi(w_1) = \varphi(w_2)$, then $\tau(\mathsf{b}_{\mathsf{w}_1}) = \tau(\mathsf{b}_{\mathsf{w}_2})$ and so $\tau(\mathsf{node}(\mathsf{bg}_{\mathsf{H}}, \mathsf{b}_{\mathsf{w}_1})) = \tau(\mathsf{node}(\mathsf{bg}_{\mathsf{H}}, \mathsf{b}_{\mathsf{w}_2}))$. But by Lemma 32, this is absurd. Finally, assume $(w_1, w_2) \in E_H$, then $\mathsf{edge}(\mathsf{b}_{\mathsf{w}_1}, \mathsf{b}_{\mathsf{w}_2}) \in \mathcal{A}_p(G,H)^{\mathsf{equi}}$. Remember that there exist $v_1, v_2 \in V_G$ such that $\tau(\mathsf{b}_{\mathsf{w}_1}) = \mathsf{c}_{\mathsf{v}_1}$ and $\tau(\mathsf{b}_{\mathsf{w}_2}) = \mathsf{c}_{\mathsf{v}_2}$; moreover, by definition, $\varphi(w_1) = v_2$ and $\varphi(w_2) = v_2$. So, since τ is an x-matching and $\mathsf{edge}(\mathsf{b}_{\mathsf{w}_1}, \mathsf{b}_{\mathsf{w}_2}) \in \mathcal{A}_p(G,H)^{\mathsf{equi}}$, we have that $\tau(\mathsf{edge}(\mathsf{b}_{\mathsf{w}_1}, \mathsf{b}_{\mathsf{w}_2})) = \mathsf{edge}(\mathsf{c}_{\varphi(\mathsf{w}_1)}, \mathsf{c}_{\varphi(\mathsf{w}_2)}) \in \mathcal{A}(G,H)$. Finally, by definition of $\mathcal{A}(G,H)$, we have that $(\varphi(w_1), \varphi(w_2)) \in E_G$. So, by Proposition 30, there exists a subgraph isomorphism of H in G. □

We use the previous lemmas to prove the NP-hardness of k-x-UNMATCHABILITY for $x \in \{\mathsf{equi}, \mathsf{poly}\}$.

Theorem 34. k-x-UNMATCHABILITY *is NP-hard for* $x \in \{\mathsf{equi}, \mathsf{poly}\}$.

Proof. We prove the theorem by reducing subgraph isomorphism to k-x-UNMATCHABILITY. We provide details for $k = 2$; proofs can be extended to $k > 2$ by including $k - 1$ encodings of G in the ABoxes.

Let G, H be graphs, we build the ABoxes $\mathcal{A}_p(G,H)^{\mathsf{equi}}$ and $\mathcal{A}(G,H)$ in polynomial time. Let $\mathcal{Q} = \{\mathsf{g_H}\}$. Now, assume there exists a subgraph isomorphism φ of H in G and let us prove that $\mathcal{A}_p(G,H)^{\mathsf{equi}}$ is 2-x-unmatchable in $\mathcal{A}(G,H)$ w.r.t. $\mathcal{Q}$. Assume by contradiction there exist a blank $b \in sig(\mathcal{A}_p(G,H)^{\mathsf{equi}}) \cap \mathsf{N_B}$ such that for each x-matching τ of $\mathcal{A}_p(G,H)^{\mathsf{equi}}$ in $\mathcal{A}(G,H)$, $\tau(b) = \mathsf{g_H}$. Now consider the x-matching τ_a, by definition of x-matching, for each $\alpha \in \mathcal{A}_p(G,H)^{\mathsf{equi}}$, $\tau_a(\alpha) \in \mathcal{A}(G,H)$. In particular, if $b \in sig(\alpha)$, then $\mathsf{g_G} \in sig(\tau(\alpha))$. So, by definition of $\mathcal{A}(G,H)$, $\tau(\alpha)$ is necessarily of type $\mathsf{graph}(\mathsf{g_H}, a)$, hence α must be of type $\mathsf{graph}(b, a')$. By definition of $\mathcal{A}_p(G,H)^{\mathsf{equi}}$, then, b is either $\mathsf{bg_H}$ or another blank bg^v_G, for some $v \in V_G$. By definition of τ_a, though, for each $v \in V_G$ the blanks bg^v_G are mapped in $\mathsf{g_G}$, and this means that $b = \mathsf{bg_H}$. Now, consider the x-matching τ_φ, by Lemma 31, $\tau_\varphi(\mathsf{bg_H}) = \mathsf{g_G}$, so we reach an absurd. Conversely, assume that $\mathcal{A}_p(G,H)^{\mathsf{equi}}$ is 2-x-unmatchable in $\mathcal{A}(G,H)$ w.r.t. $\mathcal{Q}$. Then, for each blank $b \in sig(\mathcal{A}_p(G,H)^{\mathsf{equi}}) \cap \mathsf{N_B}$, there exists an x-matching τ of $\mathcal{A}_p(G,H)^{\mathsf{equi}}$ in $\mathcal{A}(G,H)$ such that $\tau(b) \neq \mathsf{g_H}$. Thus, if we choose $b = \mathsf{bg_H}$, we can find an x-matching τ such that $\tau(\mathsf{bg_H}) \neq \mathsf{g_H}$ and so by Lemma 33, there exists a subgraph isomorphism of H in G. □

Now we focus on k-x-ANONYMIZATION. For $x = \mathsf{strict}$, the following theorem provides the same lower bound as k-strict-UNMATCHABILITY.

Theorem 35. k-strict-ANONYMIZATION *is gi-hard.*

Proof. Let G, H be graphs and consider the ABox $\mathcal{A}(G,H)$ (which can be construed in polynomial time) and let $\mathcal{Q} = sig(\mathcal{A}(G,H))$. In Theorem 17, we defined for each ABox $\mathcal{A}$, an ABox $\mathcal{A}^g$ and we proved that for each ABox $\mathcal{A}$, there exists a k-strict-unmatchable anonymization of $\mathcal{A}$ w.r.t. $\mathcal{Q}$ iff $\mathcal{A}^g$ is k-strict-unmatchable in $\mathcal{A}$ w.r.t. $\mathcal{Q}$. Now observe that in our case $\mathcal{A}(G,H)^g$ is (up to blank renaming) the ABox $\mathcal{A}_p(G,H)^{\mathsf{strict}}$. Therefore, Theorem 27 proves that the graphs G and H are isomorphic iff $\mathcal{A}_p(G,H)^{\mathsf{strict}}$ is k-strict-unmatchable in $\mathcal{A}(G,H)$ w.r.t. $\mathcal{Q}$, hence the thesis. □

Finally, for k-x-OPT-ANONYMIZATION, we have the following lower bound.

Theorem 36. k-x-OPT-ANONYMIZATION *is gi-hard, for all* $x \in \{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$.

Proof. We are proving this result by reducing the problem k-strict-UNMATCHABILITY, where $\mathcal{Q} = sig(\mathcal{A}) \setminus \mathsf{N_B}$ (k-strict-UNMATCHABILITY remains gi-hard, as the proof of Theorem 27 satisfies this restriction).

For all ABoxes $\mathcal{A}$, let $\mathcal{Q}_\mathcal{A} = sig(\mathcal{A}) \setminus \mathsf{N_B}$ and let $occ_{\mathsf{N_I}}(\mathcal{A})$ denote the number of occurrences of individuals in $\mathcal{A}$.

Transform any given instance $(\mathcal{A}, \mathcal{A}_p, \mathcal{Q}_\mathcal{A})$ of k-strict-UNMATCHABILITY into the instance $(\mathcal{A}, \mathcal{Q}_\mathcal{A}, \ell)$ of k-x-OPT-ANONYMIZATION where x is any member of $\{\mathsf{poly}, \mathsf{equi}, \mathsf{strict}\}$ and $\ell = occ_{\mathsf{N_I}}(\mathcal{A}) + |sig(\mathcal{A})|$. Note that this transformation can be computed in polynomial time.

Since $\mathcal{Q}_\mathcal{A}$ covers all constant names of $\mathcal{A}$, all x-anonymizations $\mathcal{A}_p$ of $\mathcal{A}$ satisfy $sig(\mathcal{A}_p) \subseteq \mathsf{N_B}$ (by Def. 5.(i)). Consequently, $occ_{\mathsf{N_B}}(\mathcal{A}_p) = occ_{\mathsf{N_I}}(\mathcal{A})$ and $|sig(\mathcal{A}_p) \cap \mathsf{N_B}| = |sig(\mathcal{A}_p)|$, thus $conc(\mathcal{A}_p) = occ_{\mathsf{N_I}}(\mathcal{A}) + |sig(\mathcal{A}_p)|$. In order to satisfy $conc(\mathcal{A}_p) \leq \ell$, it must be $|sig(\mathcal{A}_p)| \leq |sig(\mathcal{A})|$, which is possible only if $\mathcal{A}_p$ is a strict-anonymization. It follows immediately that $(\mathcal{A}, \mathcal{Q}_\mathcal{A}, \ell)$ is in k-x-OPT-ANONYMIZATION iff $(\mathcal{A}, \mathcal{A}_p, \mathcal{Q}_\mathcal{A})$ belongs to k-strict-UNMATCHABILITY. □

5 Related Work

Graph anonymization based on variants of k-anonymity has been investigated for social networks. Usually the attacker is assumed to know only the neighborhood of subjects, and the isomorphism of subjects required by k-anonymity is accordingly limited to adjacent nodes (Liu and Terzi 2008; Zhou and Pei 2008; Zhou and Pei 2011; Hoang, Carminati, and Ferrari 2020; Hoang, Carminati, and Ferrari 2022; Hoang, Carminati, and Ferrari 2023). Graphs are anonymized only by adding and deleting edges. These works use cost functions different from $conc$, because $conc$ has the same value in all anonymizations, due to the fact that only subjects are blanked (in some models all nodes are subjects). Only a few works prove that the counterparts of k-x-OPT-ANONYMIZATION are NP-hard (Zhou and Pei 2008; Zou, Chen, and Özsu 2009); there is no attempt at characterizing exactly the complexity of anonymization, especially upper bounds.

The graphs and anonymizations of (Cuenca Grau and Kostylev 2019) are similar to ours, however the confidentiality goal is preventing the entailment of the answers to a set of queries called policies. The works summarized in (Baader et al. 2021) adopt a similar confidentiality goal in a more general framework, comprising terminological knowledge. Since policy compliance cannot prevent Sweeney's attack, these works can be regarded as complementary to ours.

6 Conclusions and Future Work

We introduced k-unmatchability and systematically analyzed the complexity of the related decision problems for three types of anonymizations, based on replacing constants with blanks. Our results are summarized in Table 1. In most cases, we prove narrow complexity bounds comprised between gi-hard and NP. In two cases we obtained a complete characterization (NP-complete). We also found that k-x-ANONYMIZATION ranges from trivial to gi-hard as x is progressively restricted.

We plan to refine our theoretical analysis, to narrow complexity gaps. Additionally, we aim at identifying tractable cases for k-x-OPT-ANONYMIZATION and approximate solutions to the general problem, proving their scalability on real-world knowledge graphs. Furthermore, we are planning to extend our theoretical analysis along several di-

rections, such as: more general anonymizations, capable of edge removal and value abstraction; refinements of k-anonymity like *l-diversity* (Machanavajjhala et al. 2007) and *t-closeness* (Li, Li, and Venkatasubramanian 2007); and more general knowledge bases which include terminological knowledge.

Acknowledgements

The authors acknowledge financial support from the PNRR MUR project PE0000013-FAIR, the PRIN 2022 project 2022LA8XBH-POLAR, and project SERICS (PE00000014) under the NRRP MUR program funded by the EU-NGEU.

References

Baader, F.; Koopmann, P.; Kriegel, F.; Nuradiansyah, A.; and Peñaloza, R. 2021. Privacy-preserving ontology publishing: The case of quantified ABoxes w.r.t. a static cycle-restricted $\mathcal{EL}$ TBox. In Homola, M.; Ryzhikov, V.; and Schmidt, R. A., eds., *Proceedings of the 34th International Workshop on Description Logics (DL 2021) part of Bratislava Knowledge September (BAKS 2021), Bratislava, Slovakia, September 19th to 22nd, 2021*, volume 2954 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Bayardo, R. J., and Agrawal, R. 2005. Data privacy through optimal k-anonymization. In Aberer, K.; Franklin, M. J.; and Nishio, S., eds., *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*, 217–228. IEEE Computer Society.

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.

Bizer, C. 2009. The emerging web of linked data. *IEEE Intell. Syst.* 24(5):87–92.

Bonatti, P. A., and Sauro, L. 2013. A confidentiality model for ontologies. In *ISWC-13*, volume 8218 of *LNCS*, 17–32. Springer.

Cuenca Grau, B., and Kostylev, E. V. 2019. Logical foundations of linked data anonymisation. *Journal of Artificial Intelligence Research* 64:253–314.

di Vimercati, S. D. C.; Foresti, S.; Livraga, G.; and Samarati, P. 2023. k-anonymity: From theory to applications. *Trans. Data Priv.* 16(1):25–49.

Dwork, C. 2006. Differential privacy. In Bugliesi, M.; Preneel, B.; Sassone, V.; and Wegener, I., eds., *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, 1–12. Springer.

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

Hoang, A.; Carminati, B.; and Ferrari, E. 2020. Cluster-based anonymization of knowledge graphs. In Conti, M.; Zhou, J.; Casalicchio, E.; and Spognardi, A., eds., *Applied Cryptography and Network Security - 18th International Conference, ACNS 2020, Rome, Italy, October 19-22, 2020, Proceedings, Part II*, volume 12147 of *Lecture Notes in Computer Science*, 104–123. Springer.

Hoang, A.-T.; Carminati, B.; and Ferrari, E. 2022. Time-aware anonymization of knowledge graphs. *ACM Transactions on Privacy and Security*.

Hoang, A.-T.; Carminati, B.; and Ferrari, E. 2023. Protecting privacy in knowledge graphs with personalized anonymization. *IEEE Transactions on Dependable and Secure Computing* 1–12.

Li, N.; Li, T.; and Venkatasubramanian, S. 2007. t-closeness: Privacy beyond k-anonymity and l-diversity. In Chirkova, R.; Dogac, A.; Özsu, M. T.; and Sellis, T. K., eds., *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20, 2007*, 106–115. IEEE Computer Society.

Liu, K., and Terzi, E. 2008. Towards identity anonymization on graphs. In Wang, J. T., ed., *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, 93–106. ACM.

Machanavajjhala, A.; Kifer, D.; Gehrke, J.; and Venkitasubramaniam, M. 2007. *L*-diversity: Privacy beyond *k*-anonymity. *ACM Trans. Knowl. Discov. Data* 1(1):3.

Ohm, P. 2010. Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA Law Review* 57.

Sweeney, L. 2000. Simple demographics often identify people uniquely. Technical Report – Data Privacy Working Paper 3, Carnegie Mellon University, Pittsburgh.

Sweeney, L. 2002. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* 10(5):557–570.

Zhou, B., and Pei, J. 2008. Preserving privacy in social networks against neighborhood attacks. In Alonso, G.; Blakeley, J. A.; and Chen, A. L. P., eds., *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, Mexico*, 506–515. IEEE Computer Society.

Zhou, B., and Pei, J. 2011. The *k*-anonymity and *l*-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowl. Inf. Syst.* 28(1):47–77.

Zou, L.; Chen, L.; and Özsu, M. T. 2009. K-automorphism: A general framework for privacy preserving network publication. *Proc. VLDB Endow.* 2(1):946–957.