Versatile Publishing For Privacy Preservation

[Technical Report]

Xin Jin, Mingyang Zhang, Nan Zhang *
Dept. of Computer Science
George Washington University
{xjin, mingyang, nzhang10}@gwu.edu

Gautam Das †
Dept. of Computer Science and Engineering
University of Texas at Arlington
gdas@uta.edu

ABSTRACT

Motivated by the insufficiency of the existing quasi-identifier/sensitive-attribute (QI-SA) framework on modeling real-world privacy requirements for data publishing, we propose a novel *versatile publishing* scheme with which privacy requirements can be specified as an arbitrary set of privacy rules over attributes in the microdata table. To enable versatile publishing, we introduce the *Guardian Normal Form* (GNF), a novel method of publishing multiple sub-tables such that each sub-table is anonymized by an existing QI-SA publishing algorithm, while the combination of all published tables guarantees all privacy rules. We devise two algorithms, Guardian Decomposition (GD) and Utility-aware Decomposition (UAD), for decomposing a microdata table into GNF, and present extensive experiments over real-world datasets to demonstrate the effectiveness of both algorithms.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Security, Algorithms, Performance

Keywords

Privacy preservation, Versatile publishing, Guardian normal form, Decomposition

1. INTRODUCTION

Privacy-preserving data publishing (PPDP) aims to publish a *microdata* table for research and statistical analysis, without disclosing sensitive information at the individual level. A large number of such microdata tables are published regularly, especially in the healthcare industry. For example, the Texas Department of State Health Services publishes every year a table of patients discharged from more than 450 state-licensed hospitals [35]. Table 1 depicts an example of such a microdata table. We selected 6 out of 260 attributes for display. Attribute hospital denotes the ID of the hospital which reports a tuple. ICD-9-CM denotes the diagnostic code.

Table 1: A microdata table of hospital patient discharge data

hospital	age	gender	zipcode	race	ICD-9-CM
111111	37	F	71000	asian	HIV
111111	71	M	72000	white	diabetes
222222	55	F	73000	black	diabetes
222222	37	F	74000	white	flu
333333	23	M	71000	black	alcoholism
333333	37	M	72000	white	HIV

Traditionally, an attribute is called a *sensitive attribute* (SA) if it carries sensitive information at the individual level - e.g., disease, salary, etc. There is another class of attributes called *quasi-identifier* (QI). Typical QI attributes include age, zipcode, etc. These attributes may be linked (by an adversary) with external data sources (e.g., voter registry [33]) to re-identify the SA of an individual from the published table, thereby raising privacy concerns.

1.1 Motivation

Most of the existing work on PPDP [20, 23–25, 40, 45] can be considered as enforcing a single privacy rule QI $\nrightarrow$ SA - i.e., to ensure that an adversary with knowledge of QI cannot infer the SA of a tuple (beyond a pre-defined privacy guarantee such as ℓ-diversity [24]), where QI and SA are two disjoint sets of attributes in the microdata table.

The main motivation for this paper is an observation that the enforcement of a single privacy rule does not always suffice for specifying the complex privacy requirements of real-world applications. For example, the following three rules are selected out of nine[1] from the user manual of the aforementioned Texas inpatient discharge data [35]. All rules must be properly enforced before the Texas Department of State Health Services publishes the dataset.

Rule I: If a hospital has fewer than five discharges of a particular gender, then suppress[2] the zipcode of its patients of that gender.
Rule II: If the ICD-9-CM of a patient indicates HIV or alcohol/drug abuse, then suppress the gender and zipcode of that patient.
Rule III: If a hospital has fewer than ten discharges of a race, then for all of its patients of that race, change race to *other*.

*Partly supported by NSF grants 0852673, 0852674, 0845644, 0915834 and a GWU Research Enhancement Fund.
†Partly supported by NSF grants 0845644, 0812601, 0915834 and grants from Microsoft Research and Nokia Research.

KDD'10, July 25–28, 2010, Washington, DC, USA.

[1]We selected these three rules because they only involve the six attributes displayed in Table 1.
[2]Note that "suppress" should be interpreted symbolically (e.g., as being protected by a privacy guarantee) rather than literally as in the manual, because otherwise even after suppression an adversary might still be able to infer the zipcode of a patient from other attributes, such as county.

From the perspective of PPDP, the above three statements can be interpreted as the following rules[3]. Each rule has its own QI, i.e., left-hand-side (LHS) attributes, and SA, i.e., right-hand-side (RHS) attribute. Rule I, for example, states that an adversary should not be able to infer a patient's zipcode from the published data even given knowledge of the hospital and gender of that patient.

Rule I: hospital, gender $\nrightarrow$ zipcode
Rule II: ICD-9-CM $\nrightarrow$ gender
ICD-9-CM $\nrightarrow$ zipcode
Rule III: hospital $\nrightarrow$ race

One can see from this interpretation that a single privacy rule is insufficient for describing the complex privacy requirement of the Texas Health Service - instead, the published data must satisfy *multiple* privacy rules. Moreover, an attribute in the microdata table is not restricted to be either QI or SA - e.g., gender is treated as both QI in the first rule and SA in the second - an attribute can also be "neutral" (i.e., neither QI nor SA) as shown in the existing work [3].

The requirement of multiple privacy rules can be identified from many other real-world applications - e.g., the US Cancer Statistics Data published by the Center for Disease Control and Prevention is required to satisfy the following two statements [4] which we can again translate into two different privacy rules:

Rule I: Suppress count and rate when a cell has fewer than 16 individuals.

Rule II: Suppress count and rate when a cell has race attribute value limited to *"other races combined"*.

1.2 Versatile Publishing: A Novel Problem

To properly model a real-world privacy requirement, we define *versatile publishing*, a novel framework which specifies the privacy requirement of publishing a microdata table as an arbitrary set of *privacy rules*. Each rule $\{Q_1, \ldots, Q_p\} \nrightarrow \{S_1, \ldots, S_r\}$ ensures that an adversary with knowledge of all the LHS attributes $Q_1, \ldots, Q_p$ cannot learn the RHS attributes $S_1, \ldots, S_r$ beyond a pre-defined privacy guarantee[4] such as ℓ-diveristy [24], (α, k)-anonymity [39], t-closeness [20], ρ_1-to-ρ_2 breach [25, 34], etc. With this definition, most existing work on PPDP can be considered as special cases, each enforcing only one privacy rule.

For the ease of understanding, throughout the paper we use Table 1 along with the following three privacy rules as a simple running example - we shall consider more attributes and privacy rules from the Texas dataset in the experiments.

Rule 1:	age, ICD-9-CM $\nrightarrow$ race
Rule 2:	gender, ICD-9-CM $\nrightarrow$ zipcode
Rule 3:	hospital, race $\nrightarrow$ zipcode

The axioms used to infer between sets of privacy rules are subtle - which we shall discuss in §3 along with the proof of soundness and completeness. It is important to recognize that our work is transparent and orthogonal to studies on defining and achieving privacy guarantees. Indeed, any existing guarantee defined for the QI-SA framework can be readily used to define a privacy rule in versatile publishing.

1.3 Challenges to Versatile Publishing

One seemingly simple solution to versatile publishing is the direct application of the *single-table multiple-SA publishing* [24] method. For brevity, we use the term *multi-SA publishing* to refer to this approach. With multi-SA publishing, one defines as SA all attributes that appear on the RHS of at least one privacy rule, and QI as the set of all other attributes. In the running example, zipcode and race will be treated as SA, while QI will consist of age, gender, hospital and ICD-9-CM. If such a table can be properly anonymized, then the result satisfies all three privacy rules.

[3]While the interpretations are somewhat subjective, we believe that they reflect the key spirit of the above-mentioned privacy statements.

[4]We shall further discuss the semantics of multiple attributes on the RHS of a privacy rule in §2. Its precise definition is not important at this point.

Table 2: An example of publishing multiple (bucketized) tables

(a)

age	ICD-9-CM	race
37	HIV	asian
37	HIV	white
23	alcoholism	white
37	flu	black
55	diabetes	white
71	diabetes	black

(b)

gender	ICD-9-CM	zipcode
F	HIV	71000
M	HIV	72000
M	alcoholism	71000
F	flu	74000
F	diabetes	72000
M	diabetes	73000

(c)

hospital	race	zipcode
111111	asian	71000
222222	black	73000
222222	white	72000
111111	white	74000
333333	white	71000
333333	black	72000

Nonetheless, a well-understood drawback of multi-SA publishing is that it might overly reduce the utility of the published table, especially when the number of SAs increases [24]. To satisfy ℓ-diversity, for example, each anonymous group (e.g., a group of QI-indistinguishable tuples after generalization [24]) must contain at least ℓ^m tuples, where m is the number of SA attributes (e.g., m = 2 for the running example) - leading to a rapidly decreasing utility of the published data when multiple privacy rules have to be satisfied. We shall verify this observation in the experiments.

Another seemingly simple solution is to decompose the original table into a set of smaller ones for publishing, such that each small table satisfies an individual privacy rule. Table 2 shows an example of publishing three 2-diversity tables (bucketized by anatomy [40]) to satisfy Rules 1-3, respectively. The problem with this solution is that the published tables may be vulnerable to the following *intersection attack* [11, 29, 37]: First, by joining Tables 2(a) and (b), an adversary can learn that either (*asian*, 71000) or (*asian*, 72000) appears in the microdata table. Whereas, it is known from Table 2(c) that *asian* should be associated with 71000 or 73000. By *intersecting* the two conjectures, one can infer an original tuple (111111, *asian*, 71000). This violates Rule 3: {hospital, race $\nrightarrow$ zipcode}. Similarly, there are 3 other tuples that can be compromised through this intersection attack: (222222, *black*, 73000), (333333, *black*, 72000) and (333333, *white*, 71000). Such a vulnerability remains when generalization [32] is used instead of bucketization.

The intersection attack could be easily dismissed by single-attribute publishing - i.e., an extreme decomposition which publishes each attribute as an individual table. Table 1, for example, can be decomposed into 6 disjoint sub-tables. However, such a publishing method yields low utility because it destroys any correlation information between different attributes.

In summary, to the best of our knowledge, no existing or simple solution to versatile publishing can produce satisfiable results by enforcing multiple privacy rules while maintaining a reasonable level of utility for the published table(s).

1.4 Outline of Technical Results

To enable versatile publishing, we consider a decomposition of the original microdata table into multiple sub-tables with possibly overlapping attributes, such that at most one privacy rule applies to each sub-table, allowing the sub-table to be anonymized by existing PPDP algorithms under the QI-SA framework (e.g., [13,17,18,40]).

To avoid the intersection attack and to provide criteria for determining whether a privacy rule is satisfied over the multiple published tables, we develop the *Guardian Normal Form* (GNF), a normal form for the schema of published tables which guarantees that all privacy rules are satisfied over the collection of all published tables. The essence of GNF is the existence of a *guardian table* in the published schema for each privacy rule that needs to be satisfied.

Our GNF is close in spirit to normal forms in relational database

theory. In particular, like normal forms, GNF is defined over the schema of published tables, rather than the tuples in them. As a result, GNF is generic to a variety of privacy guarantees (e.g., ℓ-diversity [24] and its variants [19,39,45], t-closeness [20], ρ_1-to-ρ_2 breach [9, 34]).

Similar to database normalization, there are many different ways to decompose a microdata table into GNF. The selection of a proper decomposition should be made with the utility of the published tables under consideration. In terms of how to use the multiple published tables in GNF (e.g., for data mining), we follow the same idea as the marginal publishing technique [15] proposed for injecting utility into PPDP. In particular, each published sub-table represents a duplicate-preserving view (i.e., marginal) of the original table. To use the published tables, one needs to combine information from all published marginals to estimate a multinomial model over the original table. As in [15], we use the theory of log-linear modeling to generate a maximum likelihood estimate according to constraints given by all published marginals. A key difference between our work and [15], however, is that we publish marginals for the purpose of satisfying multiple privacy constraints, while the objective of [15] is to increase the utility of published data.

We prove that the optimization of utility for decomposing a table into GNF is NP-hard (through reduction from MIN-VERTEX-COLORING). Then, we develop two carefully designed heuristics for GNF decomposition. The first, *Guardian Decomposition* (GD), is similar to the normalization of a relational schema (e.g., into BCNF) - it identifies a privacy rule which violates GNF, and then decomposes the corresponding table into two to eliminate the violation. While the GD algorithm itself is simple and efficient, optimizing its utility can be computationally very expensive for a microdata table with a large number of attributes. To address such high-dimensional cases, we devise *Utility-Aware Decomposition* (UAD), an efficient decomposition algorithm based on a vertex-coloring heuristic.

1.5 Summary of Contributions

- We define the novel problem of *versatile publishing* which captures the real-world requirement of enforing multiple privacy rules over the publishing of a microdata table.
- We derive the *sound and complete* set of inference axioms for privacy rules in versatile publishing.
- We define *guardian normal form (GNF)* which guarantees a set of multiple privacy rules over the collection of multiple published tables.
- For decomposing a table into GNF, we prove the utility optimization to be NP-hard, and develop two heuristic algorithms GD and UAD.
- We conduct a comprehensive set of experiments over two real-world datasets, a widely-used benchmark Adult dataset [1] and the aforementioned Texas dataset. The results demonstrate the superiority of GD and UAD over the multi-SA and single-attribute publishing techniques.

The rest of the paper is organized as follows. §2 introduces preliminary notions and defines versatile publishing. The inference axioms between privacy rules are derived in §3. We describe GNF in §4 and develop the corresponding decomposition algorithms GD and UAD in §5. The experimental results are presented in §6, followed by related work in §7 and final remarks in §8.

2. VERSATILE PUBLISHING

2.1 Notations

We begin with essential notations. Let $T = \{t_1, \ldots, t_n\}$ be a microdata table of n tuples and m attributes $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$. Let $t_j[A_i]$ be the value of attribute A_i of tuple t_j. As with most existing work in PPDP, we assume all attributes to be discrete, and leave the optimal discretization of continuous attributes as a separate problem for future work.

A (deterministic or randomized) privacy-preserving data publishing algorithm perturbs T to one or more published tables based on user-specified privacy rules. We denote the set of d published tables by $\mathcal{T}^* = \{T_1^*, T_2^*, \ldots, T_d^*\}$. In general, we use the calligraphic font (e.g., $\mathcal{T}^*$ or $\mathcal{A}$) to represent a set (of tables or attributes).

2.2 Definition of Privacy Rule

Privacy requirements can be defined in various ways. A baseline definition is a *single-SA privacy rule* which specifies one attribute S as the sensitive attribute (SA). The privacy rule states that, for all tuples $t \in T$, given the published table and the QI attribute values of t, no adversary is capable of inferring $t[S]$ beyond a pre-defined guarantee. We shall discuss further details of this guarantee in the next subsection where we define the privacy measure used by versatile publishing.

In this paper, we consider a more flexible specification of privacy requirements by allowing one to define *multiple* privacy rules - e.g., a set of privacy rules $\mathcal{Q} \twoheadrightarrow S$ where $\mathcal{Q} \subseteq \mathcal{A}$, $S \in \mathcal{A}$ and $S \notin \mathcal{Q}$. Each privacy rule specifies its LHS (i.e., $\mathcal{Q}$) and RHS (i.e., S) attributes, and requires that for all tuples $t \in T$, given the published table and the LHS attribute values of t (i.e., $t[\mathcal{Q}]$), no adversary is capable of inferring the RHS attribute value of t (i.e., $t[S]$) beyond a pre-defined guarantee.

More generally, we can allow multiple attributes to be included in the RHS of a privacy rule. This way, a privacy rule $\mathcal{Q} \twoheadrightarrow \mathcal{S}$ ($\mathcal{Q} \subseteq \mathcal{A}$, $\mathcal{S} \subseteq \mathcal{A}$, $\mathcal{Q} \cap \mathcal{S} = \phi$) states that $\forall t \in T$, given the published table and $t[\mathcal{Q}]$, the *composition* of the RHS attributes of t cannot be inferred beyond a pre-defined guarantee. For example, when ℓ-diversity is used, the privacy rule requires at least ℓ well-represented value combinations for the RHS attributes. Note that such a multiple-RHS-attribute rule is *weaker* than the same rule with a subset of the RHS attributes - i.e., a publishing method which satisfies the latter automatically satisfies the former. This stands in contrast to the semantics of multiple RHS attributes in the multi-SA publishing method [24], which requires *all* SA attributes to be protected, making it *stronger* than the case with a subset of SA attributes. Our choice here is made for the sake of completeness - note that the stronger multi-SA rule can be specified as $|\mathcal{S}|$ single-SA rules where $|\mathcal{S}|$ is the number of attributes in $\mathcal{S}$, i.e., each attribute $A_i \in \mathcal{S}$ is protected by a privacy rule $\mathcal{A} \backslash A_i \twoheadrightarrow A_i$. On the other hand, the weaker privacy rules cannot be specified as a combination of multiple single-RHS-attribute rules.

Nonetheless, we do recognize that real-world privacy requirements rarely specify the composition of multiple attributes as the RHS of a privacy rule. For example, none of the privacy rules for the Texas inpatient data includes more than one RHS attributes. Hence, this paper focuses on the cases where each privacy rule has a single RHS attribute. As mentioned above, this covers the multi-SA case in the traditional sense, as the privacy requirement there can be specified as a set of single-RHS-attribute rules.

2.3 Privacy Guarantee

As discussed above, a privacy rule $\mathcal{Q} \twoheadrightarrow S$ requires that S be protected from an adversary with knowledge of $\mathcal{Q}$ and the published tables. Such a rule needs to be instantiated by a privacy guarantee with a threshold on the degree of disclosure of S. Popular existing measures that may be used include ℓ-diversity [24], (α, k)-anonymity [39], t-closeness [20], and ρ_1-to-ρ_2 breach [9, 34], etc.

Note that our focus in this paper is not to study a specific privacy measure, but to address the change from enforcing one single privacy rule to enforcing multiple rules. For this purpose, we consider a generic privacy measure defined by Kifer [6], and assume all privacy rules to adopt the same privacy guarantee (though over different attributes). In fact, our work can be easily extended to the case when the privacy guarantee specified in one privacy rule is different from another. Kifer's measure captures the difference between an adversary's belief before and after observing the published table(s). In the rest of this paper, unless specified otherwise, we use Kifer's generic privacy measure.

DEFINITION 1. *(Kifer's Generic Privacy Measure [6]) To satisfy a privacy rule $\mathcal{Q} \nrightarrow S$, for any tuple $t \in T$, an attacker's prior and posterior belief about $t[S]$, i.e., $P(t[S]|t[\mathcal{Q}])$ and $P(t[S]|t[\mathcal{Q}], \mathcal{T}^*)$, must satisfy $\delta(P(t[S]|t[\mathcal{Q}]), P(t[S]|t[\mathcal{Q}], \mathcal{T}^*)) \leq b$, where $\delta(\cdot,\cdot)$ is a distance function between two probability distributions, $\mathcal{T}^*$ is published table(s) and b is a data-publisher-specified threshold.*

In summary, the objective of versatile publishing is to publish (multiple) tables from the original microdata, such that the set of privacy rules defined by the data publisher can be satisfied simultaneously. As discussed in §1, we can consider traditional PPDP to be a special case of our setting with which the privacy requirement is specified as one privacy rule QI $\nrightarrow$ SA.

3. INFERENCE FOR PRIVACY RULES

We now consider the inference axioms for privacy rules. A privacy rule r can be inferred from a set of privacy rules $\mathcal{R}$ iff r is always satisfied when all privacy rules in $\mathcal{R}$ are satisfied. Somewhat surprisingly, we find that the only possible inference is the trivial one, that is, a privacy rule r_1 can be inferred from r_2 if the LHS attributes of r_1 is a subset of that of r_2. The following theorem shows the completeness of such a trivial inference axiom.

THEOREM 3.1. *(Completeness of Privacy Rule Inference) A privacy rule $\mathcal{Q} \nrightarrow S$ can be inferred from a set of privacy rules $\mathcal{R}$ iff there exists a rule $\mathcal{Q}' \nrightarrow S$ in $\mathcal{R}$ such that $\mathcal{Q} \subseteq \mathcal{Q}'$.*

PROOF. The correctness of the inference axiom directly follows from the definition of privacy rule. For the completeness of the axiom, we prove by contradiction. Note that we only need to construct such a contradiction for one specific instantiation of the generic privacy measure defined in Definition 1 because if an(other) axiom does not hold for this specific instantiation, then it clearly does not hold for the generic definition. In the proof, we consider the privacy measure for every rule to be ℓ-diversity with $\ell = 2$.

Suppose $\mathcal{Q} \nrightarrow S$ can be inferred from a set of privacy rules $\mathcal{R}$ which contains no rule of the form $\mathcal{Q}' \nrightarrow S$ where $\mathcal{Q} \subseteq \mathcal{Q}'$. Without loss of generality, let $\mathcal{Q} = \{A_1, \ldots, A_k\}$ ($k \in [1, m-1]$) and $S = A_m$.

In the following, we reach a contradiction by constructing a table T which satisfies every possible privacy rule $\mathcal{Q}_0 \nrightarrow S_0$ unless $Q \subseteq \mathcal{Q}_0$ and $S = S_0$. The existence of T shows that, if any other inference axiom (which cannot be derived from the trivial one) existed, then $\mathcal{Q} \nrightarrow S$ would be satisfied - this contradicts the fact that T violates $\mathcal{Q} \nrightarrow S$.

We construct such a table T as follows: Let the domain of attribute A_i be $\Omega_i = \{0, 1, a, b\}$ when $i \in [1, k]$ and $\Omega_i = \{0, 1\}$ when $i \in [k+1, m]$. There are 2^{m+k} tuples in the table. The projection of these tuples on $A_1, \ldots, A_{m-1}$ enumerate all possible value combinations for these attributes (note that the Cartesian product of $A_1, \ldots, A_{m-1}$ has size $4^k \times 2^{m-k} = 2^{m+k}$). Let A_m be

$$A_m = \text{xor}(f(A_1), \ldots, f(A_k)) \quad (1)$$

where xor is the exclusive-OR function and

$$f(x) = \begin{cases} 0, & \text{if } x \in \{0, a\}; \\ 1, & \text{if } x \in \{1, b\}, \end{cases} \quad (2)$$

for all tuples in the table.

We use three steps to prove that T satisfies every privacy rule $\mathcal{Q}_0 \nrightarrow S_0$ unless $Q \subseteq \mathcal{Q}_0$ and $S = S_0$. These three steps address three disjoint and exhaustive subsets of such privacy rules respectively. First, consider a privacy rule in which A_m does not appear. Since the values of $A_1, \ldots, A_{m-1}$ are essentially independent (i.e., they enumerate all possible value combinations), every privacy rule not involving A_m must satisfy the 2-diversity privacy guarantee.

Second, consider a privacy rule $\mathcal{Q}_0 \nrightarrow S_0$ in which $S_0 = A_m$. Since $\mathcal{Q} \not\subseteq \mathcal{Q}_0$, at least one of $A_1, \ldots, A_k$ must be absent from the LHS. According to (1), given any value combination of the LHS, there must be equal number of 0s and 1s for A_m. Thus, such a privacy rule must also satisfy the 2-diversity privacy guarantee.

Finally, consider a privacy rule $\mathcal{Q}_0 \nrightarrow S_0$ in which A_m appears on the LHS. If $S_0 \in \{A_{k+1}, \ldots, A_{m-1}\}$, 2-diversity must be satisfied because S is independent of all LHS attributes. If $S_0 \in \{A_1, \ldots, A_k\}$, since $\mathcal{Q} \not\subseteq \mathcal{Q}_0$, given any value combination for the LHS attributes, there must be equal number of tuples with $S_0 = 0$ (resp. 1) and $S_0 = a$ (resp. b). Thus, such a privacy rule must also satisfy 2-diversity. □

In fact, we can derive a similar theorem for privacy rules with more than one RHS attributes. Again, the only possible inference rules are the trivial ones:

COROLLARY 3.1.1. *A privacy rule $\mathcal{Q} \nrightarrow \mathcal{S}$ can be inferred from a set of privacy rules $\mathcal{R}$ iff there exists a rule $\mathcal{Q}' \nrightarrow \mathcal{S}'$ in $\mathcal{R}$ such that $\mathcal{Q} \subseteq \mathcal{Q}'$ and $\mathcal{S}' \subseteq \mathcal{S}$.*

Since our focus is on privacy rules with a single RHS attribute, we omit the proof in this paper.

Based on the theorem and the corollary, we can define an irreducible set of privacy rules as follows.

DEFINITION 2. *(Irreducibility of Privacy Rule Set) A set of privacy rules Ω is* irreducible *iff it does not contain two privacy rules $\mathcal{Q} \nrightarrow \mathcal{S}$ and $\mathcal{Q}' \nrightarrow \mathcal{S}'$ such that $\mathcal{Q} \subseteq \mathcal{Q}'$ and $\mathcal{S}' \subseteq \mathcal{S}$.*

For example, $\{A_1 \nrightarrow A_3, \{A_1, A_2\} \nrightarrow A_3\}$ is not irreducible because it can be reduced to $\{A_1, A_2\} \nrightarrow A_3$, which is an irreducible set. Without loss of generality, we focus on an irreducible set of privacy rules for the rest of this paper.

Connection with Functional Dependencies: Intuitively, the definition of privacy rules somewhat resembles functional dependencies (FDs), though we have shown their reduction rules to be quite different. The connections between these two concepts do not end with their definitions. In particular, when an adversary can learn certain functional dependencies through external knowledge [5,25], the privacy rules may have to be expanded to protect against additional privacy disclosure. The following theorem illustrates the case with the ℓ-diversity privacy guarantee.

THEOREM 3.2. *(Functional Dependencies in External Knowledge) For the ℓ-diversity privacy guarantee, a set of published tables satisfies $\mathcal{Q} \nrightarrow S$ only if, for any FD $\mathcal{X} \rightarrow S$ in adversarial knowledge, the published tables satisfy $\mathcal{Q} \nrightarrow \mathcal{X}$.*

PROOF. If the published tables $\mathcal{T}^*$ violate $\mathcal{Q} \nrightarrow \mathcal{X}$, then there must exist a tuple $t \in T$ and a value $v_\mathcal{X}$ in the domain of $\mathcal{X}$, such that $\Pr\{t[\mathcal{X}] = v_\mathcal{X} | t[\mathcal{Q}], \mathcal{T}^*\} > 1/\ell$. Since $\mathcal{X} \rightarrow \mathcal{S}$ is a FD, there must exist a value $v_\mathcal{S}$ in the domain of $\mathcal{S}$ such that $\Pr\{t[\mathcal{S}] = v_\mathcal{S} | t[\mathcal{X}] = v_\mathcal{X}\} = 1$. Thus, $\Pr\{t[\mathcal{S}] = v_\mathcal{S} | t[\mathcal{Q}], \mathcal{T}^*\} \geq \Pr\{t[\mathcal{X}] = v_\mathcal{X} | t[\mathcal{Q}], \mathcal{T}^*\} > 1/\ell$. One can see that $\mathcal{Q} \nrightarrow S$ is violated. □

Table 3: A GNF example of publishing multiple (bucketized) tables

(a)

hospital	age	gender	ICD-9-CM
111111	37	F	HIV
111111	71	M	diabetes
222222	37	F	flu
333333	37	M	HIV
222222	55	F	diabetes
333333	23	M	alcoholism

(b)

race	zipcode
asian	71000
white	72000
white	72000
white	74000
black	71000
black	73000

(c)

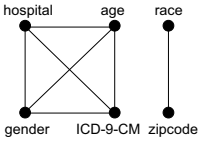
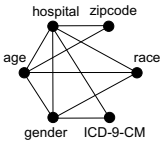


Table 4: A non-GNF example of publishing multiple tables. (a) Published table schemas (left) and enforced privacy rules (right). (b) The graphic representation of $\mathcal{T}^* = \{T_1^*, T_2^*, T_3^*, T_4^*\}$.

(a)

T_1^* = **(gender, ICD-9-CM, hospital)**	ICD-9-CM, gender $\twoheadrightarrow$ hospital
T_2^* = **(age, hospital, zipcode)**	hospital $\twoheadrightarrow$ zipcode
T_3^* = **(age, hospital, gender, race)**	age, hospital, gender $\twoheadrightarrow$ race
T_4^* = **(age, zipcode)**	no privacy rule enforced

(b)



While the theorem illustrates that privacy rules should be expanded based on FDs that are available through external knowledge, it does not offer a complete solution: In particular, it is unknown from the theorem how $\mathcal{Q} \twoheadrightarrow S$ can be satisfied if $\mathcal{Q} \rightarrow S$ happens to be a FD, because it is impossible to enforce $\mathcal{Q} \twoheadrightarrow \mathcal{Q}$ without changing the values of $\mathcal{Q}$.

Unfortunately, this is a price we have to pay for using the existing privacy guarantees such as ℓ-diversity to instantiate the privacy rules. The reason can be seen from the following example which shows that it may not be possible to eliminate an FD-incurred disclosure by adding additional privacy rules: Let there be a table of two ternary attributes $A_1, A_2 \in \{0, 1, 2\}$ over which a privacy rule $A_1 \twoheadrightarrow A_2$ needs to be enforced with ℓ-diversity guarantee and $\ell = 2$. Consider the publishing of each attribute as an individual table: $A_1 : \{0, 0, 0, 1, 1, 2\}$ and $A_2 : \{0, 2, 1, 2, 2, 1\}$. One can see that this publishing method satisfies all 2-diversity-based privacy rules that can be specified. Nonetheless, an adversary which knows that $A_1 \rightarrow A_2$ is a FD can still infer that $\langle 0, 2\rangle$, $\langle 1, 1\rangle$ and $\langle 2, 0\rangle$ must be the original tuple values. This violates $A_1 \twoheadrightarrow A_2$. We leave the defense against such an FD-incurred disclosure as an open problem for future work.

4. GUARDIAN NORMAL FORM

To enable versatile publishing, we consider the generation of multiple sub-tables with possibly overlapping attributes, such that each sub-table only addresses (at most) one privacy rule and therefore can be processed by existing PPDP algorithms designed for the QI-SA framework. This section is focused on how to design the schema of published tables such that each privacy rule remains in effect over the collection of all published tables. We defer to §5 for discussions about the utility of published tables.

4.1 Basic Ideas of GNF

It is important to understand the implications of publishing multiple tables on the enforcement of privacy rules: Due to the intersection attack [11, 29, 37], a privacy rule satisfied by two anonymized tables individually may be broken by the combination of both tables (recall the example in Table 2). Yet a privacy rule addressed by none of the published tables may be automatically satisfied due to the separation of attributes across published tables (recall the single-attribute publishing technique discussed in §1).

Thus, it is necessary to provide criteria for determining whether a privacy rule is satisfied over a collection of published tables. We define such criteria as GNF, a normal form for the schema of published tables. In the following, we discuss the basic ideas of GNF in terms of two ways for a privacy rule to be satisfied: 1) a singular case of $\langle \mathcal{Q}, S\rangle$ non-reachability; 2) a generic case of the existence of a guardian table. In the next subsection, we shall combine these two scenarios to form the definition of GNF.

Singular Case (Non-Reachability): For the ease of understanding, we use an undirected graph to represent the schema of published tables. The graph is constructed as follows: Each vertex corresponds to an attribute in the original microdata table. An edge exists between two vertices iff there is at least one published table that contains both attributes. For example, suppose that Tables 3(a) and (b) are published for our running example. Table 3(c) depicts the graph representation. One can see that if two vertices (e.g., age and race) are not reachable from each other (i.e., no path between them exists) in the graph, then the two corresponding attributes are independent given the published tables - i.e., no correlation between them is disclosed. Thus, if every attribute in $\mathcal{Q}$ is not reachable from S, then a privacy rule $\mathcal{Q} \twoheadrightarrow S$ is satisfied.

Formally, we have the following definition and lemma:

DEFINITION 3. (Reachablility) *Two attributes A_i and A_j are reachable iff there exists a sequence of published tables $T_1^*, \ldots, T_h^*$ such that T_1^* contains A_i, T_h^* contains A_j, and T_i^* shares at least one common attribute with T_{i+1}^* for all $i \in [1, h-1]$.*

LEMMA 1. *The published tables satisfy $\mathcal{Q} \twoheadrightarrow S$ if $\forall A_i \in \mathcal{Q}$, A_i and S are not reachable.*

Please refer to Appendix A.1 for the proof of Lemma 1. Consider Rules 1-2: {age, ICD-9-CM $\twoheadrightarrow$ race} and {gender, ICD-9-CM $\twoheadrightarrow$ zipcode} in the running example. According to Lemma 1, both rules are satisfied over Tables 3(a) and (b).

General Case (Guardian Table): In the nonsingular case where $\mathcal{Q}$ and S are reachable from each other, at least one of the published tables that link $\mathcal{Q}$ and S together may have to be properly anonymized in order to satisfy $\mathcal{Q} \twoheadrightarrow S$. We define the *guardian table* of a privacy rule as follows:

DEFINITION 4. (Guardian Table) *A published table T_i^* with attributes $\mathcal{A}_i^*$ is said to be the guardian table for a privacy rule $\mathcal{Q} \twoheadrightarrow S$ iff*

(i) T_i^* contains S, and

(ii) T_i^* by itself satisfies $(\mathcal{Q} \cap \mathcal{A}_i^*) \cup \mathcal{Q}^* \twoheadrightarrow S$ where $\mathcal{Q}^* = \{A_j | A_j \in \mathcal{A}_i^* \backslash S$ and A_j is reachable from at least one attribute in $\mathcal{Q}$ over $\mathcal{T}^* \backslash T_i^*\}$, and

(iii) After removing S from T_i^*, S (if it also occurs in the other tables) is no longer reachable from any attribute in $\mathcal{Q}$.

When $\mathcal{Q}$ and S are reachable from each other, the existence of a guardian table also ensures its uniqueness due to Condition (iii). The implication of Condition (iii) indicates that all paths from (any attribute in) $\mathcal{Q}$ to S must pass through edges defined by the guardian table for $\mathcal{Q} \twoheadrightarrow S$. As a result, the enforcement of the privacy rule over the guardian table also guarantees it over the collection of all published tables, as shown by the following lemma:

LEMMA 2. *The published tables satisfy $\mathcal{Q} \twoheadrightarrow S$ if there exists a guardian table for $\mathcal{Q} \twoheadrightarrow S$.*

Please refer to Appendix A.2 for the proof of Lemma 2. In the running example of publishing Tables 3(a) and (b), one can see that

Table 3(b) serves as the guardian table for Rule 3: {hospital, race $\nrightarrow$ zipcode} where $(\mathcal{Q} \cap \mathcal{A}_i^*) \cup \mathcal{Q}^* = \{\text{race}\} \cup \phi$ and $S = \text{zipcode}$. As another example, Table 4(a) illustrates the schema of four published tables, each enforcing at most one privacy rule (specified on the right). In this case, one can verify according to Definition 4 that T_3^* is the guardian table for Rule 1: {age, ICD-9-CM $\nrightarrow$ race} where $(\mathcal{Q} \cap \mathcal{A}_i^*) \cup \mathcal{Q}^* = \{\text{age}\} \cup \{\text{hospital, gender}\}$ and $S = \text{race}$. On the other hand, no published table can serve as the guardian table for Rule 2: {gender, ICD-9-CM $\nrightarrow$ zipcode}. Consider T_2^* and T_4^*, the only two tables satisfying Condition (i). Either removing zipcode from T_2^* or from T_4^* renders the same graphic representation of the four tables (after the removal) as in Table 4(b), where zipcode is still reachable from both gender and ICD-9-CM. Thus, the published tables satisfy Rule 1 but may violate Rule 2.

4.2 Definition of GNF

We are now ready to combine both scenarios to define GNF:

DEFINITION 5. (GNF) *For a given set of privacy rules $\mathcal{R}$, a set of published tables is in GNF iff for any privacy rule $\mathcal{Q} \nrightarrow S$ in $\mathcal{R}$, either $\mathcal{Q}$ and S are not reachable from each other, or there exists a published table that is the guardian table for $\mathcal{Q} \nrightarrow S$.*

The following theorem follows directly from Lemmas 1 and 2.

THEOREM 4.1. *For a given set of privacy rules, if the published tables are in GNF, then all privacy rules are satisfied.*

For the running example, the published tables specified by Table 3 are in GNF while those in Table 4 are not.

5. DECOMPOSITION INTO GNF

Given the definition of GNF, we now consider how to decompose a microdata table into GNF. Since GNF guarantees the satisfaction of all privacy rules, the focus here is on optimizing the utility of published tables. In particular, we first discuss how to utilize the multiple published tables in applications such as data mining. We then establish the hardness of utility optimization and devise GD and UAD, two decomposition algorithms on heuristic.

5.1 How to Utilize Tables Published with GNF

A main goal of PPDP is to enable analytical applications such as data mining. While the specific need of these applications can be quite different and hard to align [21], a general requirement is the knowledge of the probability distribution of the original data. In terms of the publishing method, most existing solutions in the QI-SA framework publish one anonymized view of the original data while our decomposition approach publishes multiple views. But in terms of the ultimate usage of the published data, both publishing methods produce the same - i.e., an estimation of the original data distribution based on constraints defined by the published view(s).

Thus, we adopt as the utility measure the Kullback-Leibler divergence (KL-divergence) between the original data distribution and the maximum likelihood estimation from the published views [15]. In particular, let $\vec{v}_i = \langle v_1^i, \ldots, v_m^i \rangle$ be a possible tuple value in the multi-dimensional domain of the original data, and p_i and $\tilde{p}_i$ be the probability for a tuple to take the value of $\vec{v}_i$ given the original and the estimated distributions, respectively. The KL-divergence is defined as

$$D_{\text{KL}}(p \| \tilde{p}) = \sum_i p_i \log \frac{p_i}{\tilde{p}_i}, \tag{3}$$

which is non-negative and takes the value of 0 iff $p \equiv \tilde{p}$. Since KL-divergence measures the difference in log-likelihood between the two distributions, the smaller it is, the better utility the published data is able to provide.

The problem of estimating the original distribution based on the anonymized views was studied in [15] - we follow the solution in this paper. Specifically, log-linear modeling, a popular statistics tool to model attribute associations based on contingency tables (e.g., anonymized views), is used to estimate parameters for the original multinomial distribution. Please refer to [15] for the algorithmic details. One subtle point is that the graphic representation (i.e., interaction graph [15]) of all published tables in GNF must be triangulated [16] - i.e., there are no induced cycles of length 4 or more. Otherwise, this contradicts Condition (iii) in Definition 4. Therefore, publishing schemas generated by all decomposition algorithms in this paper are decomposable [15], whereby they support a closed-form solution for the maximum likelihood estimate of the log-linear model.

5.2 Hardness of Utility Optimization

The hardness of utility optimization comes from two sources. First, since we use the existing algorithms for the QI-SA framework to anonymize each decomposed table, the overall utility is subject to the non-optimality of these algorithms. From this perspective, the utility optimization for achieving single-table ℓ-diversity has been proved to be NP-hard [26, 42]. The second source of hardness comes from the optimization of utility during the decomposition process, as shown by the following theorem.

THEOREM 5.1. *The utility optimization for decomposing a microdata table into GNF is NP-hard.*

Please refer to Appendix A.3 for the detailed proof. The main idea of the proof is a reduction from MIN-VERTEX-COLORING [12].

5.3 Algorithm GD

Similar in spirit to the database normalization algorithm which decomposes a relation into BCNF (see Algorithm 11.3 in [8]), Algorithm GD follows a simple idea: find a privacy rule which violates GNF, decompose the existing sub-tables to address the privacy rule, and continue until no more offending privacy rule exists. Specifically, if $\mathcal{Q} \nrightarrow S$ violates GNF, then we remove all occurrences of S from the existing tables, and then insert a new sub-table which consists of attributes in $\mathcal{Q} \cup \{S\}$ and enforces $\mathcal{Q} \nrightarrow S$. The details are depicted by Steps 1 to 4 in Algorithm 1.

Such a simple decomposition does not lose any attribute, but may not produce the optimal utility either - e.g., additional attributes might be added to the decomposed tables without violating GNF. To remedy this deficiency, Step 5 onwards in Algorithm 1 uses a greedy method to add attributes back to the decomposed tables. Intuitively, it inserts attributes in decreasing order of their "effectiveness" on reducing the KL-divergence. While this method produces good utility over microdata tables with a small number of attributes, as we shall show in the experiments, it does not scale well when the number of attributes is large, mainly because the computation of KL-divergence involves a time-consuming process of constructing log-linear models across multiple tables. In addition, this method cannot change the number of published tables which is determined by the decomposition step - if initially the microdata table is decomposed into many small pieces, and then one may not be able to add back many attributes without violating GNF.

5.4 Algorithm UAD

To address the problems of GD, we develop UAD by leveraging the link between utility optimization and the MIN-VERTEX-COLORING problem, as discovered in the hardness proof. In particular, consider a graph in which each vertex corresponds to an

Algorithm 1: Guardian Decomposition (GD)

Input: The microdata table T and privacy rules
Output: Published tables $\mathcal{T}^*$

1 Choose an arbitrary privacy rule and anonymize T to T^* in order to enforce it; $\mathcal{T}^* \leftarrow \{T^*\}$;
2 Find a privacy rule $\mathcal{Q} \nrightarrow S$ which violates GNF over $\mathcal{T}^*$;
3 Remove S from all tables in $\mathcal{T}^*$;
4 Create anonymized table T_1^* which has attributes $\mathcal{Q} \cup \{S\}$ and enforces $\mathcal{Q} \nrightarrow S$; Add T_1^* to $\mathcal{T}^*$; Goto 2;
5 **foreach** *pair of attribute A_i in table T and $T^* \in \mathcal{T}^*$ such that $\mathcal{T}^*$ remains in GNF after adding A_i to T^** **do**
6 Compute KL-divergence after adding A_i to T^*;
7 Find $\langle A_i, T_j^* \rangle$ with the smallest KL-divergence value; Add A_i to T_j^*;
8 Goto 5 until no such pair exists;

attribute in the microdata table, and an edge exists between two vertices iff the two corresponding attributes *appear on two different sides of a privacy rule*[5]. The key idea of UAD stems from the following observation: according to GNF, if two attributes A_i and A_j reside on two different sides of a privacy rule (i.e., connected in the graph), then A_i and A_j cannot appear in the same decomposed table unless the table enforces a privacy rule with either A_i or A_j on the RHS. Thus, the set of LHS attributes in each decomposed table forms an independent set of the graph.

Algorithm 2: Utility-Aware Decomposition (UAD)

Input: The microdata table T and privacy rules
Output: Published tables $\mathcal{T}^*$

1 Construct a directed graph $G = \langle \mathcal{V}, \mathcal{E} \rangle$ where each vertex $v_i \in \mathcal{V}$ corresponds to an attribute $A_i \in T$; Add an edge $\langle v_i, v_j \rangle$ iff there is a privacy rule with A_i and A_j on the LHS and RHS, respectively;
2 Use DSATUR to color the undirected version of G;
3 Find $\mathcal{V}_{\max}$ as the maximum set of nodes with the same color. Break tie arbitrarily;
4 Construct a table T^* with attributes $\mathcal{V}_{\max}$; Add T^* to $\mathcal{T}^*$;
5 Find $v \in \mathcal{V} \backslash \mathcal{V}_{\max}$ which has no outgoing edge to, and most incoming edge from, vertices in $\mathcal{V}_{\max}$; Break tie arbitrarily; Goto 7 if no such v exists;
6 $\mathcal{V}_{\max} \leftarrow \mathcal{V}_{\max} \cup \{v\}$. Add v to T^* and anonymize it to enforce $\mathcal{V}'_{\max} \nrightarrow v$ where $\mathcal{V}'_{\max}$ is the subset of $\mathcal{V}_{\max}$ that have edges to v;
7 Remove $\mathcal{V}_{\max}$ and all associated edges from $\mathcal{V}$;
8 Goto 2 until $\mathcal{V}$ is empty;

Since decomposition removes from the published data correlation information between attributes in different decomposed tables, we aim to generate as few such independent sets as possible. Therefore, we call DSATUR [2], a well-known heuristic algorithm for MIN-VERTEX-COLORING, to group all vertices into a small number of independent sets (i.e., colors). Then, we pick all attributes in the largest color group ($\mathcal{V}_{\max}$ in Line 3) to construct a decomposed table. We anonymize the table by adding (and enforcing a privacy rule over) a sensitive attribute v. The only condition on v is that it never appears on the LHS of a privacy rule in which an attribute in $\mathcal{V}_{\max}$ appears on the RHS (Line 5). The construction of decomposed tables is repeated until all attributes are published.

One can easily verify that tables published by UAD are in GNF and thus satisfy all privacy rules. While the algorithm is designed with utility under consideration, it is difficult to theoretically analyze the utility of published tables due to its dependency on the original data distribution. Thus, we leave utility analysis to experimental evaluations in §6.

The time complexity of UAD depends on the underlying algorithm used to anonymize each published table. When anatomy [40] is used, anonymizing a table with n tuples to ℓ-diversity takes $O(n(\log \lambda + \ell))$, where λ is the maximum domain size of an attribute. Given the complexity of DSATUR [2] being $O(m^3)$, where m is the number of attributes, UAD terminates in $O(m^4 + n(\log \lambda + \ell))$. Since $n \gg m$ for most microdata tables, the time complexity of UAD is linear to the number of tuples in the table.

[5]Note that this is a subgraph of the graph we defined in §4 and depicted in Table 3.

Table 5: The attributes and its domains in our experiment

(a) Adult dataset

attribute	domain
age	{1-6}
country	{1-2}
education	{1-7}
marital_status	{1-7}
occupation	{1-3}
income	{1-2}
sex	{1-2}

(b) Texas dataset

attribute	domain	attribute	domain
thcic	{1-323}	diag2	{1-4193}
gender	{1-2}	diag3	{1-4251}
pat_zip	{1-1581}	p_icd9	{1-1708}
county	{1-241}	hcfa_mdc	{1-24}
pat_age	{1-22}	hcfa_drg	{1-396}
race	{1-5}	apr_drg	{1-297}
ad_diag	{1-3391}	rsk_mor	{1-6}
prn_diag	{1-3488}	and_md	{1-15000}
diag1	{1-3881}	surgeon	{1-17432}

6. EXPERIMENTS

In this section, we evaluate the performance of GD and UAD for versatile publishing on two real-world datasets (i.e., Adult [1] and Texas inpatient discharge data [35]). In particular, we compare our algorithms against two baseline techniques discussed in §1: *multi-SA publishing* and *single-attribute publishing*. Please refer to Appendix B for details of our implementation of multi-SA publishing using Binary Integer Programming (BIP).

6.1 Experimental Setup

Hardware: All our experiments were conducted on a 2.6GHz Intel Core 2 Duo machine with 2GB RAM and Windows XP OS. All algorithms were implemented using C++.

Dataset 1 (Adult): We used a well-known benchmark Adult dataset [1] which contains 7 attributes and $45,222$ tuples after removing all tuples with missing values. Since our BIP implementation of multi-SA publishing is extremely time-consuming for datasets with a large number of tuples, for the purpose of comparing with multi-SA publishing, we sampled $10,000$ tuples without replacement as our testing bed, and left the scalability testing to experiments conducted on the Texas dataset.

We used two steps to pre-process the Adult dataset in order to test multi-SA publishing within a reasonable amount of time: First, we reduced the number of tuples to $10,000$ by sampling without replacement, and 2) we reduced the number of possible values of each attribute (i.e., domain size) through generalization. In particular, similar to the generalization hierarchy used in [10], we generalized attribute *age* into 6 values: {17-24}, {25-34}, {35-44}, {45-54}, {55-64} and {64-90}; attribute *country* into 2 values: US and non_US; attribute *education* into 7 values: $preschool$, $elementary$, $secondary$, $some\_colloeage$, $associted\_degree$, $university$, $post\_graduate$; and attribute *occupation* into 3 val-

ues: $white\_collar$, $blue\_collar$ and $others$. Table 5a summarizes the domain sizes of all attributes after generalization.

Dataset 2 (Texas [35]): We also tested the aforementioned Texas patient discharge data [35]. In particular, we selected 18 attributes involved in the privacy rules specified in the user manual, removed all tuples with missing values, and used the remaining $177,148$ tuples as our testbed. Table 5b describes the domain sizes of attributes in the Texas dataset.

Privacy Guarantee: We used ℓ-diversity [24] as the privacy guarantee in the experiments. Thus, a published table $\mathcal{T}^*$ satisfies a privacy rule $\mathcal{Q} \nrightarrow S$ iff $\Pr\{t[S]|t[\mathcal{Q}], \mathcal{T}^*\} \leq 1/\ell$ for all $t \in T$ [6, 13, 38, 40]. We implemented the bucketization-based anatomy algorithm [40] as the single-table anonymization subroutine invoked by GD and UAD.

Generating Privacy Rules: We randomly generated irreducible sets of privacy rules (see definition in §3) while varying two parameters: the number of privacy rules c and the number of LHS attributes lhs (recall that there is always one RHS attribute) in each privacy rule. To compare with single-table publishing, we excluded a privacy rule if it cannot even be satisfied by publishing its RHS attribute as a separate table, because such a privacy rule cannot be satisfied by single-table publishing.

Utility Measure: For the purpose of providing an intuitive observation of the utility of published tables, in addition to the KL-divergence measure discussed in §5, we also tested another common utility measure, *relative error* [40]. Relative error measures the accuracy of answering a workload of queries of the form:

SELECT COUNT(*) FROM *Dataset*
WHERE $pred(A_1), \ldots, pred(A_{qd})$

where qd is the *query dimension* and $pred(A_i)$ denotes a predicate of A_i belonging to a set of randomly selected domain values. The size of the set is captured by a parameter called *value percentage* p (i.e., the percentage of all domain values in the set). Let *Act* and *Est* be the query answer over the microdata table T and the published tables $\mathcal{T}^*$, respectively. The *relative error* is defined as $\frac{|Act-Est|}{Act}$. For each experiment, we ran a workload of $1,000$ randomly generated queries and calculated the average relative error. We varied both parameters qd and p during the generation.

6.2 Comparisons on the Adult Dataset

With the Adult dataset, we compared the utility and efficiency of all four algorithms: multi-SA, single-attribute, GD, and UAD.

Utility vs. Number of Privacy Rules: We varied the number of privacy rules c from 1 to 9 while fixing $\ell = 2$ and $lhs = 3$ for generating the privacy rules. We ran each algorithm for 50 times. For the relative error measure, we set $qd = 3$ and $p = 40\%$. Figures 1 and 2 depict the average KL-divergence and relative error for each algorithm, respectively. Note that when $c = 1$, multi-SA publishing, GD and UAD are all equivalent. For $c > 1$, our two algorithms, GD and UAD, provide nearly identical utility which outperforms both multi-SA and single-attribute publishing, according to both measures. Also note that the utility of multi-SA publishing deteriorates significantly when the number of privacy rules increases. This is because when more attributes are added as SA, multi-SA publishing has to produce buckets (or QI-indistinguishable groups) with extremely large sizes, reducing the utility of published tables.

Efficiency: Following the same setting as Figures 1 and 2, Figure 3 depicts the efficiency of UAD, GD, and multi-SA publishing over the Adult dataset. Figure 4 depicts the efficiency of UAD over the Texas dataset to demonstrate its scalability to a much larger dataset. Single-attribute publishing has essentially no overhead because it simply splits each attribute into a different table. One can see that for the same (Adult) dataset, the computational overhead of multi-SA publishing is orders of magnitude greater than that of UAD and

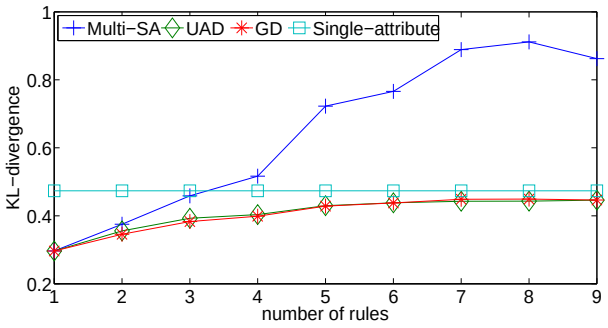


Figure 1: KL-divergence on Adult dataset, vary c

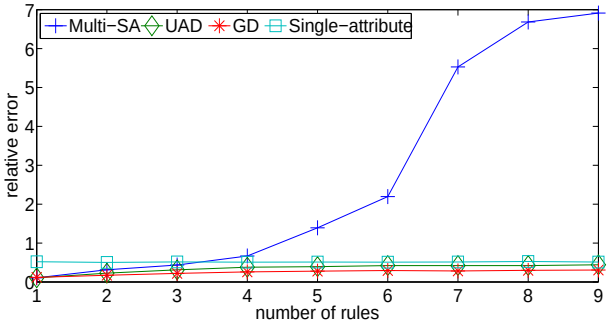


Figure 2: Relative error on Adult dataset, vary c

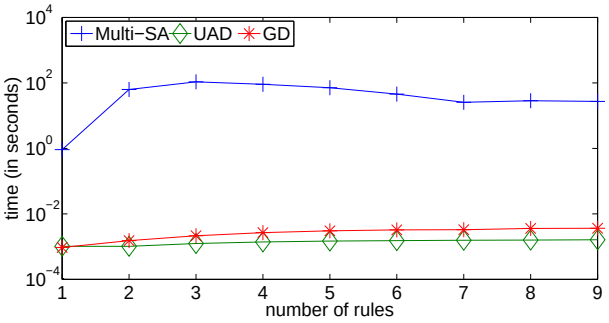


Figure 3: Running time on Adult dataset, vary c

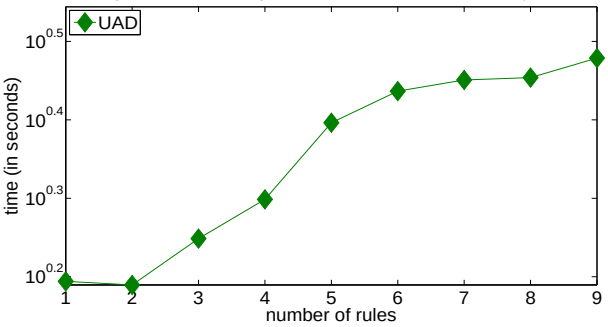


Figure 4: Running time on Texas dataset, vary c

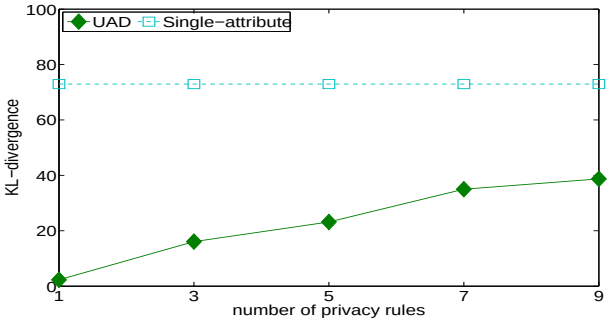


Figure 5: Kl-divergence on Texas dataset, vary c

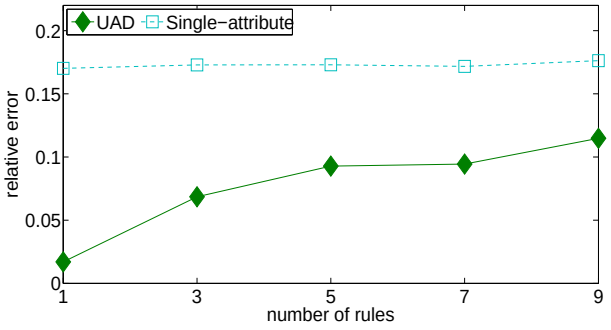


Figure 6: Relative error on Texas dataset, vary c

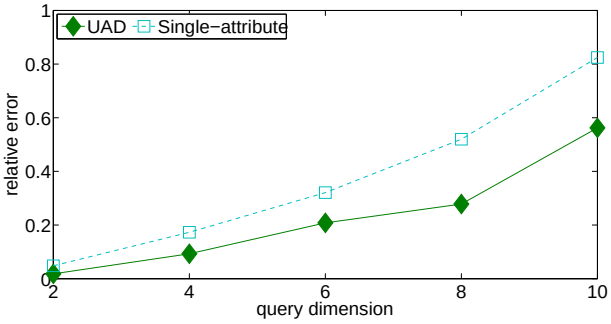


Figure 7: Relative error on Texas dataset, vary qd

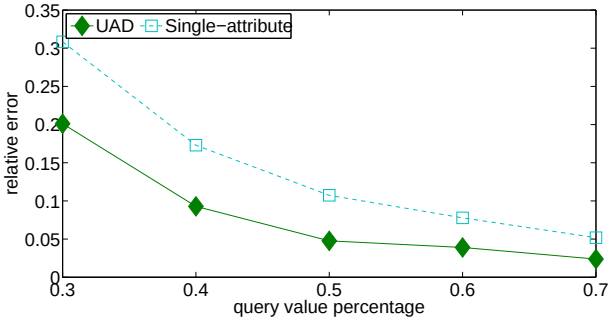


Figure 8: Relative error on Texas dataset, vary p

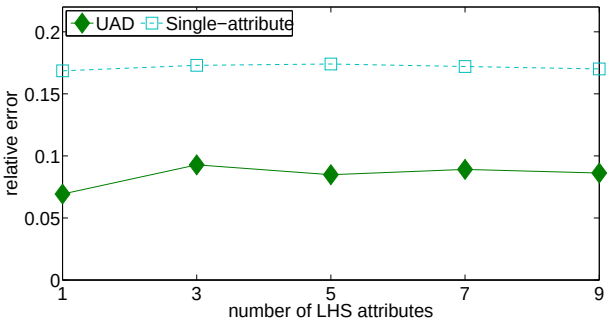


Figure 9: Texas dataset, vary lhs

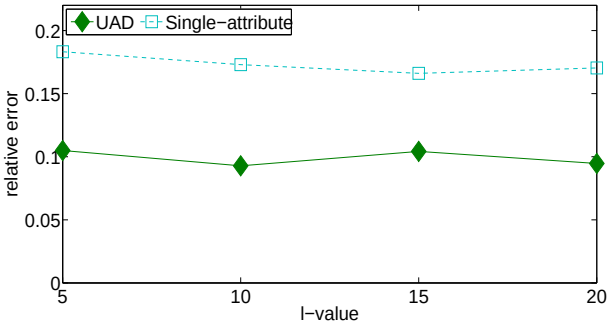


Figure 10: Relative error on Texas dataset, vary l

GD. Also note that UAD is more efficient than GD, as we explained in §5.

6.3 Comparisons on the Texas Dataset

With the Texas dataset, we focused on testing the utility of UAD and single-attribute publishing because both multi-SA publishing and GD require a matter of weeks to complete computations on the large Texas dataset.

Utility vs. Number of Privacy Rules: We set $\ell = 10$ and $lhs = 3$ for generating privacy rules and varied the number of privacy rules c from 1 to 9. For testing the relative error measure, we set $qd = 4$ and $p = 40\%$. Figure 5 and Figure 6 depict the KL-divergence and relative error of UAD and single-attribute publishing, respectively. One can see that UAD significantly outperforms single-attribute publishing for both measures.

Relative Error with Various Settings: To thoroughly test the relative error measure, we varied its settings parameters qd from 2 to 10 and p from 0.3 to 0.7. Note that we omitted $qd = 1$ because both UAD and single-attribute produce can precisely answer 1-dimensional queries. For generating privacy rules, we followed the same parameters as above except that the number of privacy rules was fixed as $c = 5$. Figures 7 and 8 depict the relative error of UAD and single-attribute publishing given the varying qd and p, respectively. Again, UAD produces smaller relative errors than single-attribute publishing under all settings.

Utility vs. Number of LHS attributes: We investigated the effect of the number of LHS attributes in each privacy rule on the utility of published tables. All other settings were kept the same as in Figure 5 with $c = 5$ and lhs varies from 1 to 9. Figure 9 depicts the results. One can see that lhs has barely any influence on the utility of UAD when $lhs \geq 3$. To understand why, consider adding an attribute A_i to the LHS of a privacy rule $\mathcal{Q} \nrightarrow S$. Given the tables published by UAD, the only situation which requires changes to the tables is when A_i and S both appear as the LHS attributes of a published table. Nonetheless, UAD is less likely to assign S to the LHS of any table, especially one that has a large number of (other) LHS attributes, because doing so prevents $\mathcal{Q}$ from being included in the table. Hence, the number of LHS attributes in a privacy rule is *not* a dominating factor for the utility of UAD.

Utility vs. ℓ: The utility of published tables is determined by two factors: (1) the decomposition of the original table, and (2) the anonymization of decomposed tables. To identify the dominating factor, we tested UAD with ℓ varying from 5 to 20 because doing so has significant impact on anonymization but no impact on decomposition. Figure 10 shows that the utility of UAD is not significantly affected by the value of ℓ. This indicates that the utility of publishing multiple tables is mainly determined by decomposition. While this paper initiates the study of decomposition design with two heuristic algorithms, we believe such importance of decomposition calls for further studies of its design in the future work.

7. RELATED WORK

Sweeney and Samarati [31, 33] first defined the k-anonymity guarantee for PPDP. After that, motivated by different sensitive information and adversarial knowledge, many other privacy guarantees have been proposed - e.g., ℓ-diversity [24], (α, k)-anonymity [39], t-closeness [20], δ-presence [28], skyline privacy [5], ρ_1-to-ρ_2 breach [9, 34], m-confidentiality [38], ϵ-privacy [23], etc. Meanwhile, various anonymization techniques, e.g., generalization and/or suppression [10, 13, 17, 18, 32], bucketization [40, 45], randomization [7, 30, 34, 41], etc, were developed to achieve these privacy guarantees. Our work is related but orthogonal to the study of these anonymization techniques. In particular, while the existing anonymization techniques mostly address PPDP at the tuple level,

we approach the problem at the schema level. As a result, our definitions of versatile publishing and GNF are transparent to the underlying privacy guarantees (e.g., ℓ-diversity). Our work is also orthogonal to the study of differential privacy [7] which, instead of providing absolute privacy guarantees on the values of sensitive attribute (like ℓ-diversity), aims to ensure minimum difference between the cases where an individual is or is not present in the database.

It is important to note that some recent work studied the *algorithm-based disclosure* of some existing anonymization techniques - i.e., disclosure that happens once an adversary learns the mechanism of the underlying annoymization algorithm [14, 22, 38, 44]. Note that such a vulnerability is transparent to our decomposition-based algorithms: Since we use the existing anonymization techniques as primitive operations, we can eliminate the threat of algorithm-based disclosure by using anonymization techniques that have been proved to be free of algorithm-based disclosure (e.g., anatomy [40], SP-Hilb [14]).

Other closely related work to ours includes the study of anonymization without explicit QI and SA attributes [37] and the protection of inference rules [36] by suppression. The generation of multiple views for optimizing utility was studied in [15], and the anonymization of a given set of views in [27, 43]. Nonetheless, both studies were designed specifically for achieving the k-anonymity guarantee. [29] addressed the publishing of a given microdata table to multiple users with different QI. While this problem can be considered as satisfying more than one privacy rules, it does not support the specification of arbitrary privacy rules with different RHS attributes as in our versatile publishing framework.

8. CONCLUSION

In this paper, we addressed the versatility problem of PPDP by defining versatile publishing which allows the specification of multiple privacy rules with arbitrary QI/SA combinations. To enable versatile publishing, we developed GNF, a normal form for publishing multiple sub-tables, each processed by an existing anonymization algorithm for a single privacy rule, to satisfy all privacy rules over the combination of all published sub-tables. To decompose a given microdata table into GNF, we devised two decomposition algorithms, GD and UAD, evaluated their performance over real-world datasets, and demonstrated their superiority over a number of baseline techniques for versatile publishing.

9. ACKNOWLEDGEMENTS

The authors would like to thank Johannes Gehrke (Cornell University) for his many insightful and helpful comments on this work, especially on the definition of privacy rules, the connection with functional dependencies, and the utilization of tables published with GNF. The authors are also grateful to the anonymous reviewers for their constructive suggestions.

10. REFERENCES

[1] A. Asuncion and D. Newman. UCI machine learning repository, 2007. `http://www.ics.uci.edu/mlearn/MLRepository.html`.

[2] D. Brelaz. New methods to color the vertices of a graph. *ACM Communication*, 22:251–256, 1979.

[3] J. Brickell and V. Shmatikov. The cost of privacy: Destruction of data-mining utility in anonymized data publishing. In *KDD*, 2008.

[4] Centers for Disease Control and Prevention. United states cancer statistics public information data 1999-2002 archive, 2009. `http://wonder.cdc.gov/`.

[5] B. Chen, R. Ramakrishnan, and K. LeFevre. Privacy skyline: Privacy with multidimensional adversarial knowledge. In *VLDB*, 2007.

[6] K. Daniel. Attacks on privacy and de finetti's theorem. In *SIGMOD*, 2009.

[7] C. Dwork. Differntial privacy. In *ICALP*, 2006.

[8] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems (4th Edition)*. Addison Wesley, 2003.

[9] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, 2003.

[10] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *ICDE*, 2005.

[11] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith. Composition attacks and auxiliary information in data privacy. In *SIGKDD*, 2008.

[12] M. Garey and D. Jonson. *Computers and Instractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, CA, 1979.

[13] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis. Fast data anonymization with low information loss. In *VLDB*, 2007.

[14] X. Jin, N. Zhang, and G. Das. Algorithm-safe privacy-preserving data publishing. In *EDBT*, 2010.

[15] D. Kifer and J. Gehrke. Injecting uility into anonymized datasets. In *SIGMOD*, pages 217–228, 2006.

[16] S. L. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

[17] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian: multidimensional k-anonymity. In *ICDE*, 2006.

[18] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: efficient full-domain k-anonymity. In *SIGMOD*, 2005.

[19] J. Li, Y. Tao, and X. Xiao. Preservation of proximity privacy in publishing numerical sensitive data. In *SIGMOD*, 2008.

[20] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *ICDE*, 2007.

[21] T. Li and N. Li. On the tradeoff between privacy and utility in data publishing. In *KDD*, 2009.

[22] W. Liu, L. Wang, and L. Zhang. k-jump strategy for privacy preserving micro-data disclosure,. In *ICDT*, 2010.

[23] A. Machanavajjhala, J. Gehrke, and M. Götz. Data publishing against relaistic adversaries. In *VLDB*, 2009.

[24] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *ICDE*, 2006.

[25] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Halpern. Worst-case background knowledge for privacy-preserving data publishing. In *ICDE*, 2007.

[26] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *PODS*, 2004.

[27] M. Nergiz, C. Clifton, and A. Nergiz. Multirelational k-anonymity. In *ICDE*, 2007.

[28] M. E. Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals from shared databases. In *SIGMOD*, 2007.

[29] J. Pei, Y. Tao, J. Li, and X. Xiao. Privacy preserving publishing on multiple quasi-identifiers. In *ICDE*, 2009.

[30] V. Rastogi, S. Hong, and D. Suciu. The boundary between privacy and utility in data publishing. In *VLDB*, 2007.

[31] P. Samarati. Protecting respondents' identities in microdata release. *TKDE*, 13(6):1010–1027, 2001.

[32] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, CMU, SRI, 1998.

[33] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.
[34] Y. Tao, X. Xiao, J. Li, and D. Zhang. On anti-corruption privacy preserving publication. In *ICDE*, 2008.
[35] Texas Department of State Health Services. User manual of texas hospital inpatient discharge public use data file, 2008. `http://www.dshs.state.tx.us/thcic/Hospitals/HospitalData.shtm`.
[36] K. Wang, B. C. M. Fung, and P. S. Yu. Template-based privacy preservation in classification problems. In *ICDM*, 2005.
[37] K. Wang, Y. Xu, A. Fu, and R. Wong. Ff-anonymity: When quasi-identifiers are missing. In *ICDE*, 2010.
[38] R. C. Wong, A. W. Fu, K. Wang, and J. Pei. Minimality attack in privacy-preserving data publishing. In *VLDB*, 2007.
[39] R. C. Wong, J. Li, A. W. Fu, and K. Wang. (α, k)-anonymity: An enhanced k-anonymity model for privacy-preserving data publishing. In *KDD*, 2006.
[40] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In *VLDB*, 2006.
[41] X. Xiao, Y. Tao, and M. Chen. Optimal random perturbation at multiple privacy levels. In *VLDB*, 2009.
[42] X. Xiao, K. Yi, and Y. Tao. The hardness and approximation algorithms for ℓ-diversity. In *EDBT*, 2010.
[43] C. Yao, X. S. Wang, and S. Jajodia. Checking for k-anonymity violation by views. In *VLDB*, 2005.
[44] L. Zhang, S. Jajodia, and A. Brodsky. Information disclosure under realistic assumptions: Privacy versus optimality. In *CCS*, 2007.
[45] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate query answering on anonymized tables. In *ICDE*, 2007.

APPENDIX

A. PROOFS

A.1 Proof of Lemma 1

LEMMA 1. *The published tables satisfy $\mathcal{Q} \nrightarrow S$ if $\forall A_i \in \mathcal{Q}$, A_i and S are not reachable.*

PROOF. Since S is not reachable from any attribute in $\mathcal{Q}$, given any tuple $t \in T$, any values v_S in the domain of S, and any value combination $v_\mathcal{Q}$ for the attributes in $\mathcal{Q}$, there is

$$\Pr\{t[S] = v_S, t[\mathcal{Q}] = v_\mathcal{Q} | \mathcal{T}^*\} = \Pr\{t[S] = v_S | \mathcal{T}^*\} \cdot \Pr\{t[\mathcal{Q}] = v_\mathcal{Q} | \mathcal{T}^*\}. \quad (4)$$

That is, $t[S]$ and $t[\mathcal{Q}]$ are conditionally independent given the published tables $\mathcal{T}^*$. Thus, the distribution of $t[S]$ satisfies

$$P\{t[S]|t[\mathcal{Q}], \mathcal{T}^*\} = P\{t[S]|\mathcal{T}^*\}. \quad (5)$$

Recall that we assume the publication of any single attribute alone does not violate the privacy rules - i.e.,

$$\delta(P\{t[S]|\mathcal{T}^*\}, P\{t[S]|t[\mathcal{Q}]\}) \leq b \quad (6)$$

where $\delta(\cdot, \cdot)$, as defined in Definition 1, is a distance function between two distributions. Thus,

$$\delta(P\{t[S]|t[\mathcal{Q}], \mathcal{T}^*\}.P\{t[S]|t[\mathcal{Q}]) \quad (7)$$
$$= \delta(P\{t[S]|\mathcal{T}^*\}, P\{t[S]|t[\mathcal{Q}]\}) \quad (8)$$
$$\leq b. \quad (9)$$

As such, the published tables satisfy $Q \nrightarrow S$. □

A.2 Proof of Lemma 2

LEMMA 2. *The published tables satisfy $\mathcal{Q} \nrightarrow S$ if there exists a guardian table for $\mathcal{Q} \nrightarrow S$.*

PROOF. Let T_i^* be the guardian table for $\mathcal{Q} \nrightarrow S$. Recall that the set of attributes in T_i^* is $\mathcal{A}_i^*$. According to the definition of guardian table, T_i^* satisfies

$$(\mathcal{Q} \cap \mathcal{A}_i^*) \cup \mathcal{Q}^* \rightarrow S \quad (10)$$

where $\mathcal{Q}^* = \{A_j | A_j \in \mathcal{A}_i^* \backslash S$ and A_j is reachable from at least one attribute in $\mathcal{Q}$ given $\mathcal{T}^* \backslash T_i^*\}$. Let $\mathcal{W} = (\mathcal{Q} \cap \mathcal{A}_i^*) \cup \mathcal{Q}^*$. In the following, we prove that the published tables $\mathcal{T}^*$ satisfy $\mathcal{Q} \cup \mathcal{W} \nrightarrow S$. Note that due to the trivial inference rule, this implies $\mathcal{Q} \nrightarrow S$.

Consider attributes in $\mathcal{Q} \backslash \mathcal{W}$. Due to the definition of $\mathcal{W}$, these attributes much be unreachable from S. According to the proof of Lemma 1, they must be conditionally independent with S given the published tables (denoted by $(\mathcal{Q} \backslash \mathcal{W}) \perp S | \mathcal{T}^*$). Thus,

$$P(t[S]|t[\mathcal{W} \cup \mathcal{Q}], \mathcal{T}^*) \quad (11)$$
$$= P(t[S]|t[\mathcal{W}], \mathcal{T}^*) \quad (12)$$
$$= P(t[S]|t[\mathcal{W}], T_i^*) \cdot \frac{P(\mathcal{T} \backslash T_i^* | T_i^*, t[S], t[\mathcal{W}])}{P(\mathcal{T} \backslash T_i^* | T_i^*, t[\mathcal{W}])}. \quad (13)$$

Since S is not reachable from $\mathcal{Q}$ after removing T_i^* from the published tables, S must be unreachable from $\mathcal{W}$ as well after the removal of T_i^*. Thus,

$$P(t[S]|\mathcal{T}, t[\mathcal{W}]) = P(t[S]|T_i^*, t[\mathcal{W}]) \quad (14)$$

That is, $t[S] \perp (\mathcal{T} \backslash T_i^*) | T_i^*, t[\mathcal{W}]$. Thus, $(\mathcal{T} \backslash T_i^*) \perp t[S] | T_i^*, t[\mathcal{W}]$ - i.e.,

$$P(\mathcal{T} \backslash T_i^* | T_i^*, t[S], t[\mathcal{W}]) = P(\mathcal{T} \backslash T_i^* | T_i^*, t[\mathcal{W}]). \quad (15)$$

According to (13), $P(t[S]|t[\mathcal{W} \cup \mathcal{Q}], \mathcal{T}^*) = P(t[S]|t[\mathcal{W}], T_i^*)$. Since $\mathcal{W} \nrightarrow S$ is satisfied by T_i^*, $(\mathcal{W} \cup \mathcal{Q}) \nrightarrow S$ must be satisfied by $\mathcal{T}^*$, the set of all published tables. □

A.3 Proof of Theorem 5.1

THEOREM 5.1. *The utility optimization for decomposing a microdata table into GNF is NP-hard.*

PROOF. We prove by constructing a polynomial-time reduction from the NP-hard Minimum Vertex Coloring (MVC) problem to utility optimization for GNF decomposition. Given a graph $G : \langle V, E \rangle$, we construct a microdata table such that each attribute A_i is corresponding to a vertex $v_i \in V$ in the graph. Consider m tuples such that each tuple t satisfies $t[A_1] = t[A_2] = \cdots = t[A_n]$. Let the privacy guarantee be ℓ-diversity with $\ell = m$. For each edge $e : (v_i, v_j) \in E$, we construct two privacy rules $v_i \nrightarrow v_j$ and $v_j \nrightarrow v_i$.

Then, given the optimal GNF decomposition of the microdata table, we construct a solution for MVC with the following procedure: First, for each decomposed table, we assign one color to all attributes but the RHS one. If an attribute has multiple colors, then we combine all of its colors into one color. We perform such color-combination process iteratively until no attribute has more than one color. After that, we assign the RHS attribute of each table a color of its own. An attribute that appears in the RHS of multiple tables is arbitrarily assigned one of the corresponding colors.

We prove the correctness of such a reduction in two steps: First, we show that the constructed solution satisfies the requirement of vertex coloring. Note that for any given color, no edge exists between vertices of that color because otherwise the corresponding decomposed table would have to satisfy a privacy rule within its LHS attributes, therefore violating the definition of GNF. Thus, the solution is a legitimate vertex coloring.

We now prove that the constructed solution is the minimum vertex coloring - there are two key observations here: One is that the enforcement of m-diversity within a decomposed table is equivalent to actually separating the table into two - one formed by its RHS attribute and the other formed by the attributes left. The other is that after such separation, if one attribute A_i appears in two tables, then we can always join the two tables together (with A_i being the join key) because the value of $t[A_i]$ is unique for each tuple t.

Thus, the utility of a decomposed schema is solely determined by the number of mutually exclusive attribute subsets not reachable from each other - i.e., the number of colors according to the above-mentioned transformation. In particular, the utility is monotonically decreasing with an increasing number of colors. Thus, the constructed vertex coloring must be the minimum one in order for the decomposition to have the optimal utility. □

B. IMPLEMENTING MULTI-SA PUBLISHING

Algorithm 3: BIP based Multi-SA algorithm

Input: A table T and privacy rules
Output: A published table T^*

1 Assign as $\mathcal{Q}$ all the attributes in T that never occur on the RHS of any privacy rules. Assign as $\mathcal{S}$ all the other attributes in T. ;
2 Let $t_1 = 1, t_2 = 1, \ldots, t_n = 1$ be n binary variables initialized to be 1. Each variable t_i corresponds to the i^{th} tuple in T where $|T| = n$. ;
3 Conduct the function BIP ( ). ;
4 **if** there is no feasible solution **then**
5 Create a new group $suppr\_group = \emptyset$;
6 **foreach** $t_i = 1$ **do**
7 Add the i^{th} tuple into $suppr\_group$. Let $t_i = 0$. ;
8 Conduct SA suppression on $suppr\_group$, and add it to T^*;
9 **return** ;
10 Create a new group $new\_group = \emptyset$;
11 **foreach** $t_i = 1$ **do**
12 Add the i^{th} tuple into new_group. Assign $t_i = 0$. ;
13 Conduct anonymization on new_group, and add it to T^*. ;
14 Go to step 2 ;

To the best of our knowledge, no multi-SA publishing algorithm has been proposed in the literature. The basic idea of our design is to follow the traditional QI/SA partitioning schema, and to group tuples based on binary-integer-programming (BIP) such that all $|\mathcal{S}|$ privacy rules: $\mathcal{A}\backslash A_i \nrightarrow A_i$ (subject to ℓ-diversity for each $A_i \in \mathcal{S}$) can be satisfied simultaneously. Algorithm 3 details the multi-SA algorithm. It first partitions each attribute of the microdata table into either QI or SA, and then, constructs a BIP framework and conducts suppression (if necessary) on groups produced from BIP.

QI/SA partition: Line 1 describes the QI/SA partition procedure. The basic idea is to maximize the number of QI in the published table due to utility concern. Since any two attributes on different sides of a privacy rule cannot both be treated as QI, we assign as QI i.e., $\mathcal{Q}$, the union of all the attributes that never appear on the RHS of any privacy rule, and let SA i.e., $\mathcal{S}$, be attributes that have appeared on the RHS.

BIP framework construction: Line 2-Line 3 constructs the BIP framework and produces groups to be anonymized. First, construct n binary variables $\{t_1, t_2, \ldots, t_n\}$ for BIP and initialize them to 1, where n is the number of tuples in T. Variable t_i corresponds to i^{th} tuple in T. When $t_i = 1$ is returned from BIP, the i^{th} tuple in T is selected to be added in the group. Otherwise, it is not included.

Second, we describe the linear constraints in the BIP framework. Let $\mathcal{S} = \{A_1, A_2, \ldots, A_{|\mathcal{S}|}\}$ be all $|\mathcal{S}|$ SA from the previous QI/SA partition step. Let $\mathcal{D}(\cdot)$ be the domain of an attribute. To simultaneously satisfy all $|\mathcal{S}|$ privacy rules: $\mathcal{A}\backslash A_i \nrightarrow A_i, \forall A_i \in \mathcal{S}$ under ℓ-diversity measure, we have to construct c constraints in BIP where $c = |\mathcal{S}| \times |\mathcal{D}(A_1)| \times \ldots \times |\mathcal{D}(A_{|\mathcal{S}|})|$.

Now, construct the objective function subject to c constraints as follows:

$$\textit{Objective function}: \quad \max \sum_{i=1}^{n} t_i \text{ subject to}$$

$$\textit{Constraint}: \quad \ell \times \sum_{t_p[A_i]=v, t_p[\mathcal{S}\backslash A_i]=\vec{v}} t_p \leq \sum_{t_q[\mathcal{S}\backslash A_i]=\vec{v}} t_q.$$

where $\forall A_i \in \mathcal{S}$ $(1 \leq i \leq |\mathcal{S}|)$, $\forall \vec{v} \in \mathcal{D}(\mathcal{S}\backslash A_i)$ and $\forall v \in \mathcal{D}(A_i)$.

Group anonymization: For each group produced from BIP, either generalization [32] or bucketization [40] techniques can be used to anonymize the group. For the fairness of comparison, we use the same bucketization technique to anonymize each group in our algorithm (Line 13). For the residue tuple (when BIP cannot obtain any feasible solutions), we do not publish their $\mathcal{S}$ attributes by suppression (Line 4-Line 9).

The BIP problem is known to be NP-hard. Branch-and-cut methods are the most important techniques for solving a large number of integer programming problems. In practice, BIP is currently implemented in many commercial and public domain software packages such as LP_SOLVER (`http://lpsolve.sourceforge.net/5.5/`) to compute feasible solutions.