July 2019

A Novel Privacy Disclosure Risk Measure and Optimizing Privacy Preserving Data Publishing Techniques

Marmar Orooji
Louisiana State University and Agricultural and Mechanical College

A NOVEL PRIVACY DISCLOSURE RISK MEASURE AND OPTIMIZING PRIVACY PRESERVING DATA PUBLISHING TECHNIQUES

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

Engineering Science

by
Marmar Orooji
B.S., Shahed University, 2013
M.S., Louisiana State University, 2017
M.S., Louisiana State University, 2019
August 2019

To my beloved husband, Kian

& to my dear parents

for their endless love and support…

ACKNOWLEDGEMENTS

Above all, I am grateful to the almighty Allah, for blessing me this far. Then, I would like to express my sincere appreciation, primarily, to my major advisor, Dr. Gerald M. Knapp. Undoubtedly, this dissertation would not have been possible without his great support, understanding, and encouragement. I appreciate him for providing me with insights toward my academic work. I want to express my gratitude to my committee members, Dr. Jianhua Chen, Dr. Jerry Trahan, and Dr. Bin Li for their valuable advice and comments throughout this work.

I gratefully acknowledge the LSU Social Research & Evaluation Center (SREC) for their financial support throughout my PhD program. I thank all of my colleagues at SREC, especially Dr. Cecile Guin (the former director), Dr. Judith Rhodes (the current director), and Dr. Sam Robison (my direct supervisor) for providing an enjoyable and friendly work environment. I would also like to thank SREC for the contribution to this research. I appreciate all my friends, too numerous to mention here, for their kindness and help. Their camaraderie made my time at LSU quite pleasant and smooth.

Finally, I thank my husband, Kian, my parents, my parents' in-law, my siblings and all my family for their ongoing support and unconditional kindness throughout this PhD career and my entire life.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

ABSTRACT vi

CHAPTER

1 INTRODUCTION 1
- 1.1 Overview 1
- 1.2 Problem Statement 3
- 1.3 Objectives 5

2 LITERATURE REVIEW 6
- 2.1 Privacy Models 6
- 2.2 Privacy Techniques 12
- 2.3 Disclosure Risk Measures 20

3 FLEXIBLE ADVERSARY DISCLOSURE RISK (FADR) MEASURE 23
- 3.1 FADR Measure 24
- 3.2 Calculation Efficiency 29
- 3.3 Experiments 35
- 3.4 Conclusions 45

4 OPTIMIZING ANONYMIZATION 47
- 4.1 Data Utility Metric 47
- 4.2 RU Generalization Algorithm 49
- 4.3 Illustrative Example 58
- 4.4 Calculating FADR Exposed by an Anonymized Dataset 61
- 4.5 Experiments 63

5 ANONYMIZED DATASET EVALUATION 72
- 5.1 ARX Data Anonymization Tool 72
- 5.2 RU Generalization Algorithm vs. ARX Average Re-identification Risk Model 76

6 CONCLUSIONS AND FUTURE WORK 87

REFERENCES 88

VITA 93

ABSTRACT

A tremendous amount of individual-level data is generated each day, with a wide variety of uses. This data often contains sensitive information about individuals, which can be disclosed by "adversaries". Even when direct identifiers such as social security numbers are masked, an adversary may be able to recognize an individual's identity for a data record by looking at the values of quasi-identifiers (QID), known as identity disclosure, or can uncover sensitive attributes (SA) about an individual through attribute disclosure. In data privacy field, multiple disclosure risk measures have been proposed. These share two drawbacks: they do not consider identity and attribute disclosure concurrently, and they make restrictive assumptions on an adversary's knowledge and disclosure target by assuming certain attributes are QIDs and SAs with clear boundary in between. In this study, we present a Flexible Adversary Disclosure Risk (FADR) measure that addresses these limitations, by presenting a single combined metric of identity and attribute disclosure, and considering all scenarios for an adversary's knowledge and disclosure targets while providing the flexibility to model a specific disclosure preference.

In addition, we employ FADR measure to develop our novel "RU Generalization" algorithm that anonymizes a sensitive dataset to be able to publish the data for public access while preserving the privacy of individuals in the dataset. The challenge is to preserve privacy without incurring excessive information loss. Our RU Generalization algorithm is a greedy heuristic algorithm, which aims at minimizing the combination of both disclosure risk and information loss, to obtain an optimized anonymized dataset.

We have conducted a set of experiments on a benchmark dataset from 1994 Census database, to evaluate both our FADR measure and RU Generalization algorithm. We have shown the robustness of our FADR measure and the effectiveness of our RU Generalization algorithm by comparing with the benchmark anonymization algorithm.

1 INTRODUCTION

1.1 Overview

A tremendous amount of data about people is generated every day, by business, healthcare, and government computer systems and by Internet of Things (IoT) devices such as cell phones and activity monitoring wristwatches. This information is useful to marketing, decision makers, and researchers, in particular individual level data (aka microdata) which can be used for detailed modeling and machine learning. However, microdata usually contains private and sensitive information about individuals and thus is considered confidential. Consequently, these datasets cannot be made freely available for public access.

For instance, Electronic Health Records (EHRs) are a significant source for medical research purposes. Because of private data on identity, demographics, and health conditions, the Health Insurance Portability and Accountability Act (HIPAA) [1] restricts access to EHR and preserves the privacy of patients in the system. Similarly, FERPA and other federal and state legislation governs privacy of sensitive datasets containing individual level data such as those from student's school enrollment, performance, and disciplinary information, Department of Correction (DOC) records, and Office of Juvenile Justice (OJJ) records. Yet this data has significant potential in helping identify problems and improve performance of services in these areas.

This raises the question of how data owners can share their data for research purposes while not violating individuals' privacy. This problem has been recently studied in depth, in two relatively close areas; *Statistical Disclosure Control (SDC)* and *Privacy Preserving Data Publishing (PPDP).* They are both sharing the same concepts in the field of data privacy, with a little difference in their methodologies.

In SDC, confidential microdata is modified such that third parties working with these data are prevented to recognize individuals in the dataset. SDC techniques include sampling, adding noise, rounding, data swapping, etc.

In PPDP area, various privacy models are proposed, each specifies a privacy requirement. A privacy requirement assures that the privacy of individuals in a confidential dataset is preserved up to a certain level. Thus, by applying privacy techniques on a confidential microdata, it satisfies a privacy requirement and then it is ready to be published for public access.

SDC and PPDP both aim at creating privacy preserved version of a confidential dataset. The output protected dataset needs to be evaluated with respect to two criteria; privacy and data utility. Evaluating the privacy of a protected dataset is done by measuring disclosure risk. Disclosure risk is a measure indicating how much the output dataset is protected in terms of individual's privacy and how much the individuals are at risk of having their confidential information disclosed. Data utility is a measure showing how much original information is lost in the output dataset due to the

changes made on the original dataset. The goal is to have an approach that minimizes the disclosure risk while maximizing the data utility.

Microdata typically includes three types of attributes of concern from a privacy perspective: *Direct Identifiers, Quasi-Identifiers (QID),* and *Sensitive Attributes (SA).*

Direct identifiers are the attributes which are unique per person and a record can be easily matched to an individual by seeing a direct identifier in that record, such as social security number, phone number, or email address. For privacy preservation, these attributes need to be removed from the data. They are often replaced by randomly assigned identifiers in order to be able to relate multiple records of individual data.

This by itself does not guarantee de-identification of the data because there might be other data fields, named *quasi-identifiers (QID),* such as birthdate, gender, and zip code, any one of which are not unique to a person, but when they are considered together, with high probability, the combination of QID field values can be used to identify individuals. This identification may occur when an "adversary" determines a person's quasi-identifiers values from publicly available data (local census data, voter lists, tax assessors, or real estate agencies, Facebook …) or personal knowledge, and uses this information to match against quasi-identifiers appearing in the confidential dataset. For example Sweeney [2] demonstrated discovering the medical record of the governor of Massachusetts from data released by the Group Insurance Commission, after obtaining the governor's QID-values from public sources. 87% of U.S. citizens can be uniquely recognized in datasets using only their birth date, gender, and 5-digit zip code [2].

Sensitive attributes (SA) contain private and confidential information about an individual. Sensitive attributes are those that PPDP is mainly concerned about protecting from association with specific individuals.

In order to preserve the privacy of people in a dataset, it is required to know what the threats to their privacy are. Three main privacy threats have been introduced: *Identity Disclosure Threat, Attribute Disclosure Threat,* and *Membership Disclosure Threat*. These threats come from an adversary who wants to disclose private information of a person, referred to as a victim.

Identity Disclosure occurs when an adversary can recognize that a record in the released dataset belongs to an individual. In this case, the adversary knows the QID values of a victim and can match those with the QIDs of the published confidential dataset and find a matching record belonging to that person.

Attribute Disclosure occurs when an adversary can link a sensitive value to an individual. Here the adversary may not precisely identify a record of a specific victim but could infer his/her sensitive values from the published data, based on the set of sensitive values associated with the group that the victim belongs to.

Membership Disclosure occurs when an adversary can determine the existence of an individual in the published dataset when the membership of the person in that dataset itself counts as private information. This means the presence or the absence of the person's record in the released dataset already reveals private information.

In PPDP area, different privacy models are proposed for thwarting these threats and we will review them in depth in Chapter 2.

1.2 Problem Statement

Most disclosure risk measures proposed in the literature address only identity disclosure attack. These risk measures are defined based on either uniqueness or re-identification. In uniqueness measures, risk is defined as the probability that the rare combination of QID values in the privacy-preserved dataset is indeed rare in the population dataset [3]. Re-identification methods estimate the number of re-identifications an adversary can obtain by matching QIDs from external knowledge against confidential dataset through record linkage algorithms [4-8]. Re-identification methods require the assumption of knowing the exact external knowledge for an adversary. Domingo-Ferrer addressed this issue by proposing the "maximum knowledge attacker model", which considers an adversary who knows the values of all QIDs in the confidential dataset about a victim [9].

Although identity disclosure risk measures have been studied in depth, very few works proposed approaches to measure attribute disclosure attacks. Some studies proposed classification accuracy as a measure of attribute disclosure after using classifiers to predict the categorical sensitive attribute values [10, 11]. Various privacy models for attribute disclosure have been proposed. However, instead of measuring the risk of attribute disclosure, they specify a Boolean condition in which the dataset is prevented from attribute disclosure if it satisfies the condition [12]. For instance, Machanavajjhala *et al.* proposed a privacy model named "ℓ-Diversity", which requires the records with similar values in their QIDs have diverse sensitive values [13]. As another privacy model, "t-closeness" requires the distribution of sensitive values in each group of records with similar QID values to be close to the overall distribution [14].

A drawback of existing identity disclosure risk measures and attribute disclosure privacy models is that they classify specific attributes as QIDs and SAs, with a clear boundary in between. This limits an adversary's external knowledge to specific QIDs and the disclosure target to specific SAs. In reality, many adversaries exist with different external knowledge and disclosure targets. A sensitive attribute for an adversary can be a disclosure target, whereas another adversary might know that attribute about a victim and use that as a QID. Some work has been done to find QIDs by defining measures such as distinct ratio and separation ratio [15]. These measures are defined based on value frequencies in different combinations of attribute such that the combinations that lead to more unique values are more likely to form QIDs.

Since a privacy preserved microdata is evaluated in terms of disclosure risk and data utility, the optimum privacy technique is the one that results in a dataset with minimum disclosure risk and maximum data utility. However, there is always a trade-off between preserving privacy and data utility. Because, the more changes we make on the confidential dataset to reduce disclosure risk, the more information the data loses and the less data utility it preserves. Thus, developing an optimum privacy technique seems to be infeasible. Nevertheless, it has been the subject of recent studies to improve privacy techniques to achieve higher data utility while not losing privacy.

In recent studies, researchers have been interested in handling multiple sensitive attributes because initially proposed privacy models just consider single SA. Extending an initial privacy model to address multiple SAs needs a modification in its definition that preserves each SA separately. Using former algorithms to employ redefined model incurs huge information loss because the privacy requirement has become stricter. Thus, new privacy techniques need to be proposed to preserve all sensitive attributes and protect data utility at the same time. The algorithms for these techniques also need to be efficient to work with a large number of SAs.

One problem with recent studies addressing multiple SAs is that proposed algorithms are evaluated based on only a few sensitive attributes (less than 10). For instance, Wang and Zhu presented a novel algorithm that can thwart different attacks to SAs but it is just limited to two SAs and extension of their work is left for future work [16].

Another challenge with multiple SAs that is not well addressed in the literature is that sensitive attributes may have different characteristics and there is no comprehensive privacy model that can consider all these features. For example, "ℓ-Diversity" is a well-known privacy model for preserving categorical SA but it doesn't work with highly skewed SAs or numerical SAs [13]. On the other hand some models like "(k, e)- anonymity" [17] or "(ε, m) - anonymity" [18] are proposed just to handle numerical sensitive attributes. Also to handle highly skewed SAs, "t-closeness" is presented [19]. In case of handling multiple sensitive attributes, Liu et al. proposed a method to handle multiple numeric sensitive attributes [20]. SLOMS is another approach for handling multiple SAs that is based on ℓ -Diversity privacy model and thereby not so appropriate for numeric SAs [21].

In addition, when the dataset contains many sensitive attributes, they may be in different levels of sensitivity, meaning that some SA may contain more sensitive information than others. For instance, consider "Disease" and "Occupation" as sensitive attributes of people in a dataset. "Disease" is considered more sensitive than "Occupation" and people are stricter in keeping their disease information personal and private compared to their occupation information.

Even within one sensitive attribute, some values can be more sensitive than others especially for binary attributes. For example, consider a sensitive attribute *Dropout Flag* which shows whether a student has dropped out of school (Y) or not (N). Having *Dropout Flag* as 'Y' is more sensitive than 'N' and in terms of preserving the privacy; it is more important to hide the identification of students who have dropped out of school.

Having different sensitivity level, either between SAs or between values within one SA, is not addressed among recent studies dealing with multiple SAs. If we consider these characteristics for our sensitive attributes, we may be able to relax some privacy requirements and therefore better preserves data utility.

1.3 Objectives

The objectives of this study are:

1. Develop Flexible Adversary Disclosure Risk (FADR) measure, as a novel disclosure risk measure which:
    a. Captures both identity and attribute disclosure attack
    b. Models all possible kinds of knowledge for adversary
    c. Considers different sensitivity levels of sensitive attributes
    d. Considers different sensitivity levels of values within one sensitive attribute
2. Develop a pruning algorithm to handle calculation efficiency of FADR measure when considering all possible kinds of knowledge for adversary and having many sensitive attributes
3. Develop an optimization algorithm to minimize both disclosure risk and information loss through generalization
4. Develop an algorithm to calculate FADR measure on a generalized dataset assuming the maximum knowledge adversary
    a. Compare the FADR and information loss measures on the generalized dataset obtained from our optimization algorithm and benchmark algorithms

2 LITERATURE REVIEW

2.1 Privacy Models

In the literature, privacy models have been classified in three basic categories with respect to the three privacy threats: models against identity-disclosure, attribute-disclosure, and membership-disclosure. A privacy model formulates privacy requirements and objectives that are accomplished by the algorithms that are also focusing on data utility objectives.

In this chapter, we introduce both fundamental and recent models within each category accompanied by the algorithms.

2.1.1 Privacy Models against Identity Disclosure

These models try to thwart identity disclosure attacks through record linkage between the published dataset and an available external dataset.

	Race	Birth	Gender	ZIP	Problem
t1	Black	1965	m	0214*	short breath
t2	Black	1965	m	0214*	chest pain
t3	Black	1965	f	0213*	hypertension
t4	Black	1965	f	0213*	hypertension
t5	Black	1964	f	0213*	obesity
t6	Black	1964	f	0213*	chest pain
t7	White	1964	m	0213*	chest pain
t8	White	1964	m	0213*	obesity
t9	White	1964	m	0213*	short breath
t10	White	1967	m	0213*	chest pain
t11	White	1967	m	0213*	chest pain

Figure 1. Example of k-anonymity, where QIDs are (Race, Birth, Gender, and ZIP) and k=2 [2]

An older but fundamental model is called k-anonymity [2]. K-anonymity can be counted as a baseline for later models. It prevents record linkage as any individual can be matched with at least k records in the published dataset based on their quasi-identifiers. This is achieved by transforming quasi-identifiers and creating groups of at least k records with equal quasi-identifiers called equivalence classes. It limits the probability of identifying an individual in the published dataset down to 1/K. For instance, Figure 1 illustrates a 2-anonymity data table with race, birth year, gender, and zip code as QIDs and it contains 5 equivalence classes in which QID values are the same. Here, zip code is the only QID that is generalized by hiding its last digit. According to this anonymized data table, if an adversary wants to disclose the information of a victim who is a black

female, born in 1964, and lives in 02137 zip code, he can find 2 records having this characteristics and therefore cannot find the exact record belonging to that victim.

Since k-anonymity is originally defined for single-table datasets, one of its extensions is multi-relational k-anonymity which supports anonymization in multi-relational database schema [22]. However, it has been shown that using single-table k-anonymity algorithms for multiple relation setting either fails in protecting identity disclosure, or excessively reduces data utility of the anonymized dataset. For this model, the definition of quasi-identifier and k-anonymity are modified, and specific data utility measures are proposed to fit multi-relational setting.

Transactional data or so-called "set-valued" data are treated differently in terms of preserving privacy. Purchased items for a customer or diagnosis codes for a patient are examples of this kind of data. The privacy threat for these data occurs when an adversary has some knowledge about an individual's subset of transactional data. Terrovitis et al. have proposed K^m anonymity model as an extension of basic k-anonymity model that avoids the association of a specific transaction to a particular person [23]. Since the knowledge of the adversary is not known by the data publisher, K^m anonymity model assumes that the maximum knowledge of an adversary is at most m items of a transaction and therefore it enforces the anonymization by requiring each set of m or less items to appear in at least k records of the released dataset.

He and Naughton addressed the limitations of K^m anonymity [24]. They stated that the choice of safe m for K^m anonymity is sometimes impossible. Because it may happen that based on the background knowledge of an adversary about a victim, extra items other than those m items in a transaction cannot be matched with the victim; therefore, less than k records will be remained for being linked to the victim and that increases the risk of identity disclosure. Thus, in response to these drawbacks, He and Naughton used basic k-anonymity model instead, i.e., for any transaction there should be at least $k-1$ other identical transactions in the released dataset.

2.1.2 Privacy Models against Attribute Disclosure

These models prevent sensitive attribute disclosure for an individual. As described in Chapter 1, thwarting identity disclosure does not guarantee preventing attribute disclosure. Sometimes you may find multiple records matching an individual (like a k-anonymity dataset), therefore, you can claim that the identity disclosure for that person is prevented. However, it is possible that among those multiple matching records (equivalence class), sensitive attributes have unique values. In this case, regardless of knowing which record in the equivalence class belongs to that individual, the sensitive value is revealed and this is where attribute disclosure occurs. For instance, back in Figure 1, assume that *Problem* is the sensitive attribute. If a victim is a black female, born in 1965, and lives in the 02138 zip code, then although the adversary is finding 2 matched records for this victim, he will find that the victim has *hypertension* problem and therefore the private information of the victim is revealed. This is called homogeneity attack on k-anonymity, which leads to sensitive attribute disclosure. There is also another attack on k-anonymity model, called background knowledge attack, which can disclose the sensitive attribute of an individual by

excluding from the equivalence class those sensitive attributes that are not probable to be associated with that individual based on the background knowledge of the adversary.

Machanavajjhala et al. described these two attacks and proposed a new model, named "ℓ-Diversity", against attribute disclosure in order to thwart the aforementioned attacks and address the shortcomings of k-anonymity [13]. ℓ-Diversity requires each equivalence class to contain at least ℓ ''well represented'' sensitive attribute (SA) values. The simplest interpretation of ''well represented'' is distinct, and leads to "Distinct ℓ-Diversity", which enforces the equivalence class to have at least ℓ distinct SA values. Some variations of ℓ-Diversity with respect to the interpretation of "well represented" are as follows:

Entropy ℓ -Diversity, in which for every equivalence class E;

$$-\sum_{s\in S} p(E,s)\log\big(p(E,s)\big) \geq \log(\ell) \tag{1}$$

where $p(E,s)$ is the fraction of records in E that have the sensitive value s and S is SA domain. This criteria actually enforces that each equivalence class not only must have enough different sensitive values, but also the different sensitive values must be distributed evenly enough.

- Recursive (c,ℓ)-Diversity, which requires each class to contain a large number of distinct SA values, none of which appears too often or too rare.
- Positive Disclosure-Recursive (c, ℓ)-Diversity and Negative/Positive Disclosure-Recursive (c, ℓ)-Diversity, which capture background knowledge of an adversary and consider the cases in which some values of SA do not require protection since they are too frequent or they may not pose a threat to privacy.

There are other similar models to ℓ-Diversity, such as the "p-sensitive k-anonymity" model [25] in which parameter p acts like ℓ in Distinct ℓ -Diversity, and the "(α,k)-anonymity" model [26] in which the frequency (fraction) of a sensitive value in each equivalence class is no more than α. Both of these models enforce k-anonymity at the beginning to create equivalence classes and then protect SA in those classes. Although these models seem similar, they differ in their applied algorithms and we will introduce these algorithms in the "Privacy Techniques" section of this chapter.

A year after ℓ-Diversity was introduced, Ninghui et al. addressed two attacks on ℓ-Diversity: skewness attack and similarity attack, both of which can cause disclosure of the sensitive value(s) for an individual [19]. They proved that if the overall distribution of the sensitive attribute values is highly skewed (skewness attack) and also known to the adversary, or if the sensitive values within each equivalence class are distinct but semantically close to each other (similarity attack), then ℓ-Diversity cannot protect sensitive attributes. They generalized the background knowledge attack by replacing the prior belief of the adversary about an individual's SA with the global background knowledge that is the distribution of SA in the whole population. Therefore, they

proposed a new privacy model called "t-closeness" in which the distance between the distribution of a sensitive attribute in each equivalence class and the distribution of the attribute in the whole table is no more than a threshold *t*. They used Earth Mover's Distance (EMD) [27] to compute the distance between distributions.

Recently, Soria-Comas et al. proposed a new study on t-closeness model in which they used a different privacy technique for creating equivalence classes than the one in the original model [14]. In fact, they used a micro-aggregation technique instead of generalization in order to create k-anonymous data and apply the t-closeness model, and they proved that changing generalization to micro-aggregation improves the data utility. We will explain these techniques in the "Privacy Techniques" section.

Later, Ninghui et al. extended their proposed model to a more flexible model called "(n,t)-closeness" in order to achieve a better balance between privacy and utility [28]. In this model, instead of considering the sensitive values in the whole population, it limits the amount of sensitive information about the individuals by looking through a group with minimum size of 'n'. In fact, it enforces the distribution of any equivalence class to be close to the distribution of at least one superset of that equivalence class containing at least n records, with respect to the sensitive attribute. The other novelty of this work is that they addressed the limitation of EMD for computing the distance between distributions and therefore proposed a novel distance measure based on kernel smoothing that satisfies all of the required properties.

Thwarting similarity attacks is the subject of recent studies. This attack may be applied against either categorical or numerical sensitive attributes. Following we discuss this issue in more depth, along with the recent related works for both categorical and numeric sensitive attributes.

Similarity Attack on Numeric Sensitive Attribute

When the sensitive attribute is numeric, having diverse sensitive values is not sufficient for preventing attribute disclosure attack. Although the sensitive values are distinct, they all may fall into a short interval. For example, if a sensitive attribute is 'salary', by looking at the anonymized released table the adversary may find that an individual's salary (say, "Mary"), can possibly be \$10k, \$11k, \$13k, or \$15k. Although the adversary will not know the exact value of Mary's salary, they will find that it is within the range of \$10k and \$15k, which is a short enough interval to determine that Mary has a low income and thereby threatens her privacy. In the literature, this issue is referred to as a similarity attack on a numeric sensitive attribute, a proximity breach, or a range disclosure.

One of the foundational models dealing with proximity breach was proposed by Zhang et al. called "(k,e)- Anonymity" [17]. This model restricts each equivalence class to have at least k different sensitive values with a range of at least e. One drawback of the (k,e)-Anonymity model is that it doesn't consider the distribution of sensitive values within a range in an equivalence class. Thus, regardless of having a wide range of values, if some sensitive values occur frequently within that

range, the adversary can still find that an individual is more likely to be linked to the more frequent values within that range.

The following year, another model, "(ε, m)-anonymity" was proposed to address the aforementioned limitation of the (k,e)- Anonymity model [29]. (ε, m)-anonymity holds that, given an equivalence class E, for every sensitive value x in E at most 1/m of its tuples can have sensitive values similar to x. Being similar to x is quantified by parameter ε. For instance, two values are similar if their absolute difference is at most ε. Later, Li et al. proposed a more effective algorithm to achieve (ε, m)-anonymity than the one originally proposed in terms of gaining better data utility and less computation time [18].

Loukides et al. addressed some new issues on range disclosure attack which had not been solved in prior work [30]. Although the proposed approach considered numeric sensitive attribute, it can also be applied to the categorical sensitive attribute as well. Their method also introduced a privacy measure, called Range Diversity that allows anonymizers to specify detailed protection requirements for sensitive ranges, and quantifies the amount of protection for ranges by taking both positive and negative disclosure into account. This approach measured the probability of disclosing any range in the least protected equivalence class of a table, and captures the way sensitive attribute values form ranges in a class, based on their frequency and similarity. Through their experiments, the authors also showed that their approach achieved significantly lower data utility loss than the (ε, m)-anonymity approach by measuring data utility for the same runs of the algorithms using two different criteria for data utility metrics: Worst Group Utility (WGU) [31] and Average Utility (AU) [32].

An issue that is not covered in the above studies is the problem of having multiple numeric sensitive attributes and trying to protect the set from similarity attack. Liu et al. proposed a method to address this issue [33]. Their method uses the appropriate privacy techniques such as clustering and multi-sensitive bucketization (MSB). However, this paper does not present an algorithm to achieve this method.

Similarity Attack on Categorical Sensitive Attribute

Similarity attack on categorical sensitive attribute is also known as semantic attack. This is the case when sensitive values in an equivalence class are distinct but semantically similar. For instance, an equivalence class has 'gastric ulcer', 'gastritis', and 'stomach cancer' as distinct sensitive values for 'disease' sensitive attribute. Although these are distinct, they are semantically related and if an individual is linked to this class, an adversary will know that he has stomach-related disease.

The prior studies on thwarting similarity attack, focused on numeric sensitive attributes and their proposed models do not work for categorical sensitive attribute. However, Wang et al. proposed "(k, ε)- Anonymity" model based on the semantic similarity to thwarting similarity attack [34]. This model requires that each equivalence class in anonymous dataset satisfy k-anonymity

constraints and at the same time, any two sensitive values in the same equivalence class are not ε -similar. The definition of the ε –similar is based on the semantic hierarchical tree of a sensitive attribute. According to this approach, semantic similarity between two values can be measured by the path length between the two values on this tree.

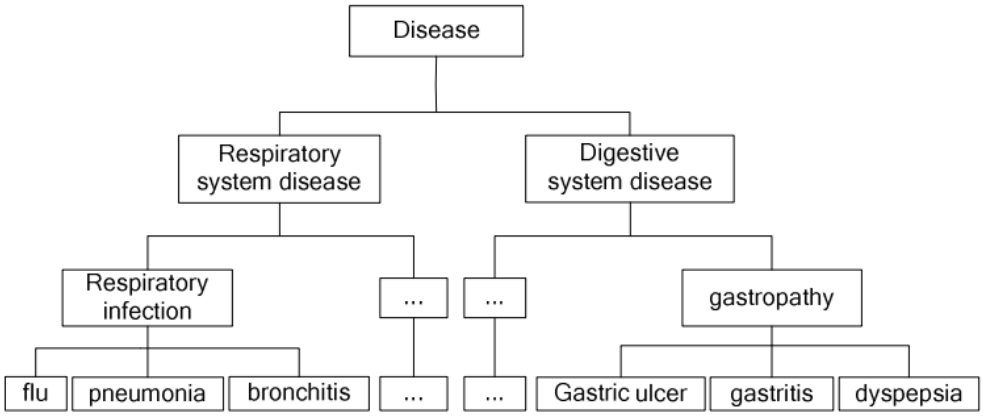


Figure 2. The semantic hierarchical tree for a disease attribute [34]

For example, Figure 2 is a semantic hierarchical tree of the disease attribute. As shown in this tree, 'gastric ulcer' and 'gastritis' have a common parent on the tree, so they are semantically similar.

However, 'flu' and 'gastritis' have only a common great grandparent, so they are comparatively dissimilar. Semantic distance is defined such that two sensitive values s_1 and s_2 are ε-similar if:

$$\frac{(h_1 - h_c) + (h_2 - h_c)}{2} \geq \varepsilon \quad (2)$$

where h_1 and h_2 are the level of s_1 and s_2 in the semantic hierarchical tree and h_c is the level of their closest common ancestor. In Figure 2, 'gastric ulcer' and 'gastritis' are 1-similar and 'flu' and 'gastritis' are 3-similar.

2.1.3 Privacy Models against Membership Disclosure

Knowing the existence of an individual in a dataset can pose a privacy risk. Therefore, here we will present the models in the literature that attempt to thwart attacks aimed at identifying the existence of an individual in an anonymized dataset.

Nergiz et al. proposed the first model called "δ-presence" [35]. δ-presence is a metric to evaluate the risk of identifying an individual in a table based on generalization of publicly known data. This model assures that the membership disclosure is protected when the probability of inferring that an individual's record is contained in a sensitive database is within a range (δmin, δmax) of acceptable probabilities. The parameters δmin and δmax are specified by data publisher who also

need to possess the population table P. P is assumed to contain ‘‘all publicly known data’’ (i.e., the direct identifiers and quasi-identifiers of all individuals in the population, including those in D).

δ-presence has a drawback that requires data owners to have access to complete information about the population, in the form of table P. Thus, the authors tried to overcome this limitation and improve their method by presenting a modified version of δ-presence, which is “c-Confident δ-presence” [36]. “c-Confident δ-presence” assumes a set of distribution functions for the population (i.e., attackers know the probability that an individual is associated with one or more values, over one or more attributes) instead of table P, and ensures that a record is δ-present with respect to the population with an owner-specified probability c.

2.2 Privacy Techniques

A common theme among privacy models is creating equivalence classes. This initially came from the “k-anonymity” model that forces creation of equivalence classes by generalizing quasi-identifiers in a way that all records in one equivalence class have the same values of quasi-identifiers. Later on, this became a privacy technique used for employing most of the privacy models either for identity-disclosure prevention or for attribute-disclosure prevention. The prevention of identity disclosure requires transforming quasi-identifiers in order to create equivalence classes in a way that it achieves privacy model requirement and data utility objectives as well. The later ensures that preserving the data privacy will not make data lose excessive information. Since transforming the data to achieve privacy and optimal utility is computationally infeasible, most algorithms adopt heuristic strategies to explore the space of possible solutions, i.e. they consider different ways of transforming quasi-identifiers in order to find a ‘‘good’’ solution that satisfies privacy and the utility objective. Therefore, the algorithms to employ privacy models usually consist of data transformation, data utility measure, and heuristic strategies to search for the “good” solution.

In the following section, we introduce transformation methods, utility objectives, and heuristic strategies addressed in the literature, and then go through the algorithms using these techniques.

2.2.1 Anonymization Operations

Transforming Quasi-Identifiers

There are three main anonymization operations for transforming QIDs of the similar records to be in an equivalence class: Generalization, Suppression, and Micro-aggregation.

Fung et al. presented different forms of generalization and suppression methods as anonymization operations in their recent survey [37].

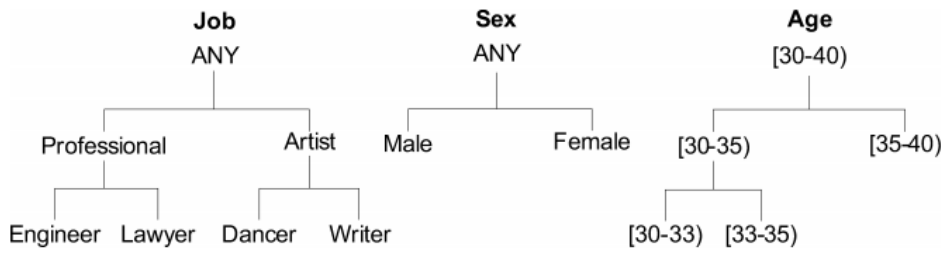


Figure 3. Generalization Hierarchy for Job, Sex, and Age [37].

The most common method found in the literature is generalization. Generalization replaces quasi-identifier's value by more general, but semantically consistent, values. Each QID has a generalization hierarchy tree, called taxonomy tree, which shows the values of the attribute as the leaves of the tree and the parent nodes are the generalized values. Figure 3, shows the taxonomy trees for QIDs *Job*, *Sex*, and *Age*.

The two main models of generalization are "global recoding" and "local recoding". In global recoding, all values of a quasi-identifier will be generalized to a same level. In contrast, local recoding generalizes those values differently; some instances may not be even generalized. For example, for local recoding, in one partition, *age* can be generalized to 10 years interval while in the other partition it is generalized to 5 years interval. But for global recoding, all partitions have age interval of 10 years. Therefore, compared with global recoding, local recoding is more flexible, and therefore it produces a smaller information loss. However, this flexibility may cause data exploration problems. For example, two instances may be treated differently in a data-mining task since they are generalized differently.

Suppression also appears in different formats: Record suppression refers to suppressing an entire record; Value suppression refers to suppressing every instance of a given value in a table; and Cell suppression (or local suppression) refers to suppressing specific instances of a given value in a table.

It is not only possible but also common for the algorithms to apply both generalization and suppression at the same time. For example, Loukides et al. employ suppression when it is not possible to apply generalization while satisfying some utility requirements [38].

Gkoulalas-Divanis et al. also added a "Micro-aggregation" method to the above transformation techniques [39]. This involves replacing a group of values in a quasi-identifier, using a summary statistic (e.g., centroid or median for numerical and categorical QIDs, respectively). This approach may harm data truthfulness, i.e., the centroid may not appear in the data.

Permutation-Based

Permutation-based approaches do not make any change to the values of the quasi-identifiers. In fact, they leave them intact to preserve more information and instead they break the association

between quasi-identifier attributes and sensitive attribute by permuting the sensitive attribute in order to prevent attribute disclosure. Therefore, you cannot certainly link any record to one sensitive value since that value is permuted and is not the original value for that person.

One method used for this kind of operation is "Bucketization". In bucketization, the step of creating equivalence classes remains the same as generalization but within each class, called a bucket, instead of generalizing the quasi-identifier's values, it separates the sensitive values from quasi-identifiers by randomly permuting the sensitive values within each bucket [40].

Zhang et al. compared generalization to permutation on the same partitioning and found that since in permutation QID values are remained intact, aggregate query answering is more accurate on permuted data than generalized data [17].

2.2.2 Data Utility Objectives

Satisfying privacy constraints based on the privacy model is one side of PPDP algorithms. The other side is retaining information so that the published data remains practically useful. There are three broad categories of information metrics for measuring data usefulness: General Purpose Metrics, Specific Purpose Metrics, and Trade-off Metrics.

General Purpose Metrics

In many cases, the data publisher does not know how the published data will be used and analyzed by the recipient. Therefore, they generally compare the anonymous data with the original data and quantify information loss incurred by data transformation in terms of an optimization measure, which they attempt to minimize. Gkoulalas-Divanis et al. classified these metrics into the following two groups [39]:

1. Metrics that look at the size of equivalence class:

   Since the records within an equivalence class share the same values over quasi-identifier, they become indistinguishable from one another and therefore if each equivalence class contains many records, that means we have high information loss. Examples of these metrics include:

   - Discernibility Metric (DM): charging a penalty to each record for being indistinguishable from other records with respect to QID.

$$C_{DM} = \sum_{Equivalence\ classes\ E} |E|^2$$

   - Normalized Average Equivalence Class Size:

$$C_{AVG} = \left(\frac{total\ \#\ of\ records}{total\ \#\ of\ equivalence\ classes}\right)/(k)$$

The drawback of this group of metrics is that they ignore the way values are transformed within each class while more generalized values lose more information compared with less generalized values. This motivates next group of metrics.

2. Metrics that penalize more generalized values include [39]

    - Minimal Distortion (MD): each level of generalization is assigned 1 unit of distortion for each value. For example, for attribute *Job*, looking at Figure 3, if 10 records with *Engineer* value generalized to *Professional*, 10 units of distortion occurs (1 unit for each record). While, if these records are generalized to *Any*, 20 units of distortion occurs (2 units for each record) because of having 2 levels of generalization.
    - Normalized Certainty Penalty (NCP): charging a penalty of $|v_g\ |/|A|$ to each generalized instance value (v_g) of attribute *A*. $\left|v_g\right|$ is the number of leaf nodes in the generalization hierarchy of A that are descendants of v_g. $|A|$ is the total number of domain values of the attribute *A*. For example, for one instance of the generalized value *Artist* in attribute *Job*, the penalty is $2/4 = 0.5$. NCP for a record is derived as the summation of NCP of all attribute values of that record and finally NCP for a dataset is the summation of NCP of all records.

Specific Purpose Metrics

If we know the tasks the data will be used for, we can take those tasks into account during anonymization to better retain information. Gkoulalas-Divanis et al. mentioned two instances of data usage tasks and their proposed information metric in the literature [39]:

- Data Classification task: The proposed metric is Classification Metric (CM) which looks at the number of records whose class labels are different from that of the majority of records in their equivalence class, normalized by the dataset size.
- Aggregate Query Answering tasks: The proposed metric is Average Relative Error (ARE) that measures the difference between the answers to a query using the anonymized dataset and the original dataset.

Trade-off Metrics

In the two aforementioned types of metrics, we look at data utility objective apart from satisfying privacy model. This means that we choose an anonymized dataset that preserves the most information. However, trade-off metrics consider both privacy and information requirements at the same time of every anonymization operation and determine an optimal trade-off between the two requirements. For instance, a metric proposed in the literature is the ratio between information gain and the privacy loos [38]. It aims at maximizing information gain per each loos of privacy. The definition of information gain and privacy loos depends on the information metric and privacy model.

2.2.3 Heuristic Strategies

The problem of creating a dataset satisfying a privacy model's requirement while optimally preserving the data utility objectives is NP-hard. As mentioned before, optimally anonymizing data with respect to the aforementioned data utility criteria is computationally infeasible**.** Consequently, heuristic strategies are employed in the algorithms to find sets of equivalence classes heuristically. In a survey studied by Gkoulalas-Divanis et al. a good classification of these strategies are presented in the following subsections [39].

Searching Strategies

These strategies are applied when using generalization to transform quasi-identifiers and create equivalence classes. They create a generalization hierarchy, called taxonomy, for each quasi-identifier attribute separately and then combine those taxonomies for all quasi-identifier attributes, to obtain a lattice. Thus, finding a way to generalize values can be performed by exploring the lattice using heuristics that avoid considering certain lattice nodes for efficiency reasons. Here are examples of heuristic lattice search methods [39]:

- Binary lattice search
- Apriori-like lattice search
- Genetic lattice search

Binary lattice search prunes the ascendants of lattice nodes that are sufficient to satisfy a privacy model while Apriori-like and Genetic lattice search prune lattice nodes that are likely to incur high utility loss.

Binary and Apriori-like lattice search strategies explore a small space of potential solutions and thus may fail to preserve data utility to the extent that genetic search strategies can do. However, genetic search is computationally intensive. Thus, more recent studies have focused on grouping strategies.

Grouping Strategies

As opposed to searching strategies, grouping strategies work on the records not the quasi-identifier values. They split the records into groups iteratively in a way to find a “good” anonymized dataset heuristically with respect to the privacy and utility. Here are the examples of heuristic grouping strategies:

- Data Partitioning: Chooses a quasi-identifier to split the records into two groups based on the values on that attribute. Each group will have similar values with respect to that quasi-identifier the splitting was based on. This method iteratively does the splitting within each group in order to get a satisfactory dataset.
- Data Clustering: In contrast to partitioning, clustering merges two groups of records based on the values of all quasi-identifier attributes together.

- Space Mapping: It ranks the records based on the values of quasi-identifiers. In fact, records with similar values in quasi-identifiers have similar ranks. It then records which consecutive ranks will form a group by satisfying privacy and utility requirements.

Gkoulalas-Divanis et al. compared partitioning-based methods with clustering-based methods and concluded that partitioning-based methods incur higher utility loss and perform poorly when the dataset is skewed and also they are sensitive to the choice of the splitting attribute. However, it is worth noting that partitioning is faster than clustering by orders of magnitude, requiring O(n.log(n)) time instead of O(n2), where n is the cardinality of the dataset. The authors also pointed that space-mapping techniques achieve good efficiency, as the ranking can be calculated in linear time, as well as being effective at preserving data utility.

2.2.4 Algorithms

In this section, we introduce some algorithms for employing privacy models while preserving data utility. These algorithms are demonstrated in

Table 1, showing their satisfied privacy model, the anonymization operation used, the data utility metric, and also the applied heuristic strategy to come up with the optimal solution.

Incognito was initially proposed by LeFevre et al. for efficiently employing k-anonymity model [41]. It searches through a lattice of all possible global recoding generalizations of quasi-identifiers and tries to find the one with the minimum distortion (MD), i.e., the one with least generalized values. The enforced searching strategy is Apriori-like method that uses Breadth-First Search algorithm and aims at efficient searching by utilizing *monotonicity property* of k-anonymity that reduces the searching space. *Monotonicity property* is saying that if a table is k-anonymous, then every generalization of that table is also k-anonymous.

When the ℓ-Diversity model was proposed, the suggested algorithm to produce optimal ℓ-Diverse data was Incognito as well. Machanavajjhala et al. stated that ℓ-Diversity also possesses the monotonicity property, i.e., if a table is ℓ-Diverse then every generalization of that table is also ℓ-Diverse, and therefore they employed their model by using Incognito algorithm in the same way as it was used for k-anonymity except that in their work Incognito satisfied ℓ-Diversity privacy constraints [13].

Similarly, t-closeness was initially employed by Incognito. Li et al. could prove that if a table satisfies t-closeness, any further generalized version of that table also satisfies t-closeness. Again this implies monotonicity property and motivates the implementation of Incognito algorithm [19].

Mondrian multidimensional partitioning algorithm enforces the k-anonymity model by recursively greedy partitioning the space based on a selected quasi-identifier. In each iteration of the algorithm, the data will be partitioned into two roughly even sized regions and the algorithm is recursively called for each of the two created regions and continues till no more partitions can be

created according to the k-anonymity constraint. In this study, the data utility is measured by Discernibility Metric and also by the specific purpose metric which focuses on answering aggregated queries accurately [42].

Table 1. Algorithms for employing privacy models and preserving data utility

Algorithm	Privacy Model	Anonymization Operation	Data Utility Metric	Heuristic Strategy
Incognito	K-anonymity & ℓ -Diversity & t-closeness	Generalization & Suppression	Minimal Distortion (MD)	Apriori-like lattice search
Mondrian	K-anonymity	Generalization	DM & Aggregate Query Answering	Data Partitioning
Mondrian	t-closeness	Generalization	NCP	Data Partitioning
LSD Mondrian	K-anonymity	Generalization	Regression Accuracy	Data Partitioning
Infogain Mondrian	K-anonymity	Generalization	Classification Accuracy	Data Partitioning
KACA	K-anonymity	Generalization	Minimal Distortion (MD)	Data Clustering
Hilbert & iDistance	K-anonymity & ℓ -Diversity	Generalization	Normalized Certainty Penalty (NCP)	Space Mapping
t-Closeness-First Microaggregation	t-closeness	Microaggregation	Sum of Squared Error (SSE)	Data Clustering
SPALM / MPALM	δ-Presence	Generalization	Minimal Distortion (MD)	Top-down lattice search
SFALM	c-Confident δ-Presence	Generalization	Minimal Distortion (MD)	Top-down lattice search
Slicing	K-anonymity & ℓ -Diversity	Bucketization	DM & Classification Accuracy	Data Partitioning

This Mondrian algorithm was extended to be adopted for different class of workload-aware anonymizations. The initial algorithm was referred as “Median Mondrian” and then “Infogain

Mondrian" and "LSD Mondrian" were also proposed as the extensions, each for specific workload [43]. For single target classification model, Infogain Mondrian is introduced in which heuristic partitioning scheme will be based on information gain and at each recursive step, the algorithm chooses a split which not only satisfies the anonymity criteria but also minimizes the weighted entropy over the set of resulting partitions. At the end it will produce homogenous partitions of class labels. For single target regression model, LSD Mondrian is proposed which recursively chooses the split that minimizes the weighted sum of mean squared errors over the set of resulting partitions.

Mondrian was also used in employing t-closeness model and its extension, (n,t)-closeness model [28]. It is also showed that (n,t)-closeness better preserves data utility than basic t-closeness model.

KACA algorithm tries to find an arbitrary equivalence class of size smaller than *k* and merge it with the closest equivalence classes to form a larger equivalence class with the smallest distortion. This process repeats recursively until each equivalent class contains at least *k* tuples [44].

The use of space mapping techniques in algorithms Hilbert and iDistance enables them to preserve data utility equally well or even better than the Mondrian algorithm. To map the space of quasi-identifiers, Hilbert algorithm uses the Hilbert curve, which can preserve the locality of points (i.e., values in quasi-identifiers) fairly well. The intuition behind using this curve is that, with high probability, two records with similar values in quasi-identifiers will also be similar with respect to their rank that is produced based on the curve. The iDistance algorithm measures similarity based on sampling and clustering of points, and is shown to be slightly inferior to Hilbert in terms of data utility [45].

Soria-Comas et al. proposed and evaluated microaggregation based algorithms to generate k-anonymous t-close data sets. They modified the microaggregation algorithm for it to take t-closeness into account at the moment of cluster formation, in an attempt to improve the utility of the anonymised data set [14].

δ-Presence model was employed by two algorithms, SPALM and MPALM. They both took advantage of anti-monotonicity property of δ-Presence, which says if a generalized table is not δ-Present, the less generalized table that locates below that table in the generalization lattice is not δ-Present either. This leads to pruning search space in top down lattice search. These two algorithms differ in the way they generalize quasi-identifiers, SPALM does global generalization while MPALM locally generalizes the values [35]. The extended model, c-Confident δ-Presence, uses similar algorithm, SFALM which is modified version of SPALM that accepts a confidence threshold and a public distribution instead of a public table [36].

Slicing is a new algorithm that addresses many issues at the same time. It addressed the limitation of generalization, such as high information loss, losing the correlation between quasi-identifiers, and curse of dimensionality. Therefore, it uses bucketization to prevent facing these drawbacks. However, it also pointed some limitations for bucketization as well, like not preventing

membership disclosure and losing the correlation between quasi-identifiers and sensitive attribute. Therefore, Slicing modified the bucketization approach by merging highly correlated attributes into columns and then apply bucketization on those columns. In this way, it is showing that the data utility is highly preserved and also membership disclosure is also prevented [40]. This approach is also extended to "Overlapping Slicing" which allows creating overlapping columns that have some attributes in common. This extension came for enhancing the data utility by preserving more correlations [46].

2.3 Disclosure Risk Measures

Thus far, we have reviewed privacy models introduced in the literature as well as privacy techniques proposed to satisfy the privacy requirements of the models. Privacy requirements specified by privacy models are parametric. The parameters express the privacy level and they can be tuned to preserve privacy of the individuals in a confidential data, up to a desired level. As we have discussed, the more we preserve individual's privacy, the more we lose data utility. Privacy models are compared one to another with respect to the information loss they incur.

Torra stated that privacy models can be seen as Boolean condition for disclosure [12]. It means the privacy requirement is like a Boolean condition for disclosure; when a confidential data satisfies the requirement, it implies that there is no disclosure risk. Therefore, in the literature, the works proposing privacy models do not measure disclosure risk after applying the model on the dataset.

However, in the literature, some disclosure risk measures have been introduced to evaluate privacy techniques that are not counted as Boolean condition for disclosure. Besides privacy models, in the literature, there is an area named *Statistical Disclosure Control (SDC)*, in which various techniques have been proposed to modify confidential microdata in order to limit disclosure. After applying these techniques to a confidential microdata and create a privacy protected dataset, they are evaluated based on a disclosure risk measure calculated on the protected dataset. Several statistical disclosure control techniques are such as sampling, adding noise, rounding, data swapping, etc.

Disclosure risk measures introduced in the literature are classified based on the type of disclosure: identity disclosure or attribute disclosure. Most works have focused on identity disclosure risk while some considered attribute disclosure risk.

2.3.1 Identity Disclosure Risk Measures

For measuring identity disclosure risk, we need to specify key attributes of the dataset. Key attributes are the same as quasi-identifiers; they are attributes which incur identity disclosure because they might be known for individuals from public data sources.

There are two types of identity disclosure risk measures; *Uniqueness* and *Re-identification*.

Uniqueness

This measure is less common than Re-identification measure because it is only used when the statistical disclosure control technique is sampling. Sampling is one of the SDC techniques which reduces disclosure risk by selecting only a subset of records from the initial microdata. The original confidential dataset is the population dataset and the privacy protected dataset is the sample dataset. Based on Uniqueness measure, identity disclosure risk is defined as the probability that the rare combination of key attribute values in the sample dataset is indeed rare in the population dataset.

Re-identification

In this type of risk measure, we assume an adversary has access to an external dataset containing direct identifiers and some quasi-identifiers for individuals. Having this external knowledge, the adversary wants to link the victim's quasi-identifiers to quasi-identifiers in the protected dataset in order to find a matched record and disclose the confidential information of the victim from the protected dataset. Thus, re-identification occurs when an adversary could identify a victim's record in the protected dataset by matching it with his external knowledge based on the key attribute values. Consequently, this type of risk measure, estimates the number of re-identifications that an adversary can obtain.

Re-identification is done through record linkage algorithms which are either probabilistic-based or distance-based.

Probabilistic-based record linkage algorithms assign weights to each pair of records in the original and protected dataset, indicating the likelihood that the two records referring to the same individual. Then, pairs with weights higher than a specified threshold are labeled as "linked". Finally, disclosure risk measure is the percentage of records in the privacy protected dataset which are labeled as "linked".

Distance-based record linkage algorithms compute distances between records in the original dataset and the protected dataset. For every record in the protected dataset, the nearest record in original dataset is marked as "linked". Disclosure risk measure is then defined as the percentage of records marked as "linked" in the protected dataset. This type of algorithms are compute-intensive and thus might not be applicable for large datasets.

One limitation of re-identification risk measure is that making assumptions about an adversary's exact external knowledge is a difficult task and a data publisher cannot perfectly model adversary's background knowledge. Domingo-Ferrer addressed this issue and proposed a "maximum knowledge attacker model" [9]. This model assumes that the adversary has access to the maximum information about individuals and therefore considers the worst-case scenario for disclosure. The maximum information about individuals that an adversary can access and incur the maximum disclosure risk, is the original confidential dataset itself. In fact, this model, assumes that the

adversary knows all the original key attribute values of the individuals in the confidential dataset, and use this information for applying re-identification and measuring disclosure risk.

2.3.2 Attribute Disclosure Risk Measures

Attribute disclosure risk measures are less studied in the literature compared to identity disclosure risk measures, because the privacy models used, such as l-diversity or t-closeness are not measurable in terms of disclosure risk; they are just Boolean conditions.

Nin et al. proposed an attribute disclosure risk measure for categorical sensitive attributes [10]. Their proposed approach is to build a classifier and use privacy protected dataset as the training dataset to predict the sensitive attribute value. Then the original confidential dataset is used as a testing dataset. Finally, the percentage of original records that are correctly classified will be considered as an estimation of the attribute disclosure risk. In other words, the accuracy of the classifier would be the measure of attribute disclosure risk.

3 FLEXIBLE ADVERSARY DISCLOSURE RISK (FADR) MEASURE

In this chapter, we develop a generalized privacy disclosure risk measure, called FADR (Flexible Adversary Disclosure Risk) measure, at record level, which considers both identity and attribute disclosure attack concurrently. We define FADR measure as the product of likelihood and consequence estimators. The likelihood of a record shows how probable the record is to be re-identified by any adversary. The consequence is then measured in terms of the sensitivity level of the information to be revealed for the record after re-identification. Thus, likelihood is a measure of identity disclosure and consequence represents attribute disclosure.

FADR measure considers all possible scenarios for an adversary's external knowledge and disclosure target by counting any subsets of attributes to be known or unknown by an adversary. Instead of restricting the adversary's knowledge to one set of attributes as QIDs, and the adversary's disclosure target to one set of attributes as SAs, we iteratively split attributes into two sets of known and unknown attributes to consider any combination of attributes once to be known and once to be unknown. The attributes in the known set act as QIDs and the ones in the unknowns set can be counted as SAs.

Moreover, our approach gives the data publisher the flexibility to assign high weight to attributes which are more probable to be publicly known about individuals in the underlying dataset as well as the flexibility to as-sign high weight to attributes which contain more sensitive information about individuals. This weighting makes known sets containing more probable attributes and unknown sets comprising more sensitive attributes, have higher impact on FADR measure.

To handle computation complexity, we proposed an efficient algorithm to prune the branches of known and unknown sets that have low contribution in FADR measure.

In summary, our contributions in this chapter include:

- Presenting FADR measure as a single combined metric of disclosure risk, which measures both identity and attribute disclosure concurrently.
- Considering all possible scenarios for an adversary's external knowledge and disclosure targets by iteratively splitting attributes into two sets of known and unknown sets.
- Providing the flexibility to data publisher for weighing high probable or sensitive attributes to have higher impact on FADR measure.
- Handling the computation complexity of FADR measure including large number of known and unknown sets, by proposing a pruning algorithm that removes sets with low contribution in risk calculation.

3.1 FADR Measure

FADR measure is defined based on risk assessment methodology. In risk assessment, risk contains two components multiplied with each other-i.e., likelihood and consequence. FADR measure is defined likewise for each record *r* in the dataset.

3.1.1 Single Combined Metric for Measuring Identity & Attribute Disclosure

In FADR measure, we use likelihood as a measure of identity disclosure and consequence for measuring at-tribute disclosure. Thus, FADR, as a single combined disclosure risk measure, is defined as:

$$FADR(r) = L(r) \times C(r), \tag{3}$$

where $FADR(r)$ is the disclosure risk, $L(r)$ likelihood, and $C(r)$ the consequence of a record *r*, respectively. $L(r)$ indicates the likelihood of record r to be re-identified by an adversary. $C(r)$ specifies that given *r* is re-identified, what is the sensitivity of the private information of *r* being revealed?

3.1.2 Considering All Possible Scenarios for an Adversary's External Knowledge and Disclosure Targets

FADR measure considers all possible scenarios for an adversary's external knowledge and disclosure target by counting any subset of attributes to be known or unknown by an adversary. We iteratively split attributes into known and unknown attribute sets. A known set contains attributes, which an adversary knows about a victim (QIDs). The remaining attributes form the unknown set and are attributes, which may contain private information (SAs) an adversary wants to disclose about a victim. The unknown set is a complement set of the known set. We consider all possible attributes' splitting. Thus, we allow an attribute to appear in a known set of a split and in an unknown set of another split. Therefore, we count all possible scenarios for an adversary's external knowledge (known set) and disclosure targets (unknown set). Since each attribute has 2 possibilities − i.e., being in the known set or in the unknown set – given *m* number of attributes, the total number of known/unknown sets is equal to 2^m. This number includes two cases of having all attributes as a known set (empty unknown set) and all attributes as an unknown set (empty known set). The two cases incur no disclosure since there is no disclosure target in the former case and no external knowledge to be used to find victims in the latter case. Excluding the two cases from all possible scenarios, FADR measure, defined in Eq. (3), is then calculated over all $2^m - 2$ sets, and is extended as:

$$FADR(r) = \sum_{i=1}^{2^m-2} L_{KS_i}(r) \times C_{UKS_i}(r), \tag{4}$$

where KS_i is the i^{th} known set of attributes and UKS_i is the i^{th} unknown set of attributes. UKS_i is the complement set of KS_i. Likelihood is calculated based on the known set and consequence

is derived from the unknown set. Thus, $L_{KS_i}(r)$ is the likelihood of the record *r* being re-identified through attributes in the i^{th} known set, and $C_{UKS_i}(r)$ is the consequence of the record *r* being re-identified in terms of the amount of private information in the i^{th} unknown set being disclosed for this record.

3.1.3 Weighting Highly Probable Known Sets

As described in Section 3.1.2, FADR measure provides the opportunity for any subset of attributes to be known by an adversary, which results in counting $2^m - 2$ number of known sets. Although an adversary with any external knowledge may exist, in practice, some subsets of attributes are more probable to be publicly known than others. Therefore, in our approach, we give the data publisher the flexibility to assign probability to each attribute, indicating how probable the attribute is to be publicly known about individuals in the underlying dataset. To make our pruning algorithm (present in section 3.2) tractable and for the ease of use by users of our measure, it is assumed that publicly known probability of each attribute is independent from one another. Thus, publicly known probability of the i^{th} known set, shown is Eq. (5), is computed as the multiplication of the publicly known probabilities of the attributes in the i^{th} known set and the publicly unknown probabilities (compliment probability) of the attributes which are not in the i^{th} known set, i.e., they exist in the i^{th} unknown set. In our terminology, publicly known and unknown probability are noted as PK and PUK, respectively.

$$PK(KS_i) = \prod_{A_j \in KS_i} PK(A_j) \times \prod_{A_k \in UKS_i} PUK(A_k) \tag{5}$$

$PK(A_j)$ is the publicly known probability of A_j, $PUK(A_k)$ is the publicly unknown probability of A_k, A_j is the j^{th} attribute in the i^{th} known set (KS_i) and A_k is the k^{th} attribute in the i^{th} unknown set (UKS_i). $PUK(A_k)$ is the complement probability of $PK(A_k)$, and thereby computed as:

$$PUK(A_k) = 1 - PK(A_k). \tag{6}$$

We use $PK(KS_i)$ in calculating $L_{KS_i}(r)$. However, KS_i can be very likely to be publicly known while KS_i's attribute values for the record r might occur frequently in the dataset. Thus, an adversary will find several matched records for r and consequently the likelihood of identity disclosure for r is decreased. Therefore, we also need to consider the frequency of KS_i's attribute values for the record r, in formulating likelihood in FADR measure. Hence, $L_{KS_i}(r)$ is derived as

$$L_{KS_i}(r) = PK(KS_i) \times 1/count(r[KS_i]), \tag{7}$$

where $PK(KS_i)$ is the publicly known probability of the i^{th} known set derived from Eq. (5), $r[KS_i]$ is i^{th} known set's attribute values for the record r, and $count(r[KS_i])$ is the number of occurrences of $r[KS_i]$ in the dataset.

Known sets comprised of attributes with higher values of PK increase the first term in Eq. (7), and therefore lead to higher likelihood of identity disclosure, compared to other known sets containing attributes with lower values of PK. This makes highly probable known sets have higher impact on FADR measure compared to low probable known sets, since FADR measure is calculated over all possible known sets. The data publisher has the flexibility to apply this impact on FADR measure by assigning PK to each attribute.

The second term in Eq. (7) considers the values of the known set of attributes for the record r. Then it counts the number of occurrences of those values together in the whole dataset. If this count is large, it means a large number of records have these values and the person whom record r belongs to is less likely to be re-identified by those attributes. Therefore, the count is inversely correlated with the likelihood. Including this term, likelihood is no longer a probability function. However, for each record it is still a value between 0 and 1.

One of the well-known re-identification risks used in the literature is *prosecutor risk* [47, 48]. This only measures identity disclosure, by considering specific attributes as QIDs. For a record r, based on a specific attributes as QIDs, prosecutor risk is measured as the inverse frequency of QID-values of the record r in the dataset, as shown in Eq. (8).

$$prosecutor\ risk_{QID}(r) = 1/count(r[\text{QID}]) \tag{8}$$

Theorem 1. $L_{KS_i}(r) = prosecutor\ risk_{QID}(r)$, if the attributes in KS_i are QID attributes and $\forall A_j \in KS_i: PK(A_j) = 1$ and $\forall A_k \in UKS_i: PK(A_k) = 0$.

Proof. Based on Eq. (6), $\forall A_k \in UKS_i: PUK(A_k) = 1 - 0 = 1$. Therefore, PK of known attributes and PUK of unknown attributes are 1. Thus, following Eq. (5), $PK(KS_i) = 1$. Substituting in likelihood formula shown in Eq. (7), likelihood of record r based on KS_i is simplified to $1/count(r[KS_i])$. Since we assumed KS_i contains QID attributes, $KS_i = QID$ and thereby $L_{KS_i}(r) = 1/count(r[\text{QID}]) = prosecutor\ risk_{QID}(r)$. □

3.1.4 Weighting Sensitive Unknown Sets

The consequence term in FADR measure considers attribute disclosure attack. After finding how likely the record r is to be identified, we measure how much private information of record r is revealed. Referring to section 3.1.2, any subset of attributes can be appeared in an unknown set to consider all possible scenarios for an adversary's disclosure target. However, attributes can be of different levels of sensitivity, depending on how much private information they hold and how much individuals are sensitive about those information.

For example, attribute *disease* is more sensitive than attribute *occupation* - people are typically stricter in keeping their disease information personal and private compared to their occupation information. Even within an attribute, some values can be more sensitive than the others. For

instance, within the values of attribute *disease*, *cancer* is of the higher severity level compared to *flu*, and thus likely to be of higher privacy concern.

Considering different sensitivity levels provides better modeling of attribute disclosure attack. Thus, in our approach, we give data publisher the flexibility to assign sensitivity weights to both attributes and their values in the underlying dataset. The attribute sensitivity weights must be integer values between 0 and 100, and the value sensitivity weights must be integer values between 0 and 1. The larger values imply higher sensitivity and incur higher consequence values in our risk measure.

Having sensitivity weights assigned for attributes and their values, we define $C_{UKS_i}(r)$ as:

$$C_{UKS_i}(r) = \sum_{A_j \in UKS_i} \left(SW(A_j) \times SW(r[A_j]) \right), \tag{9}$$

where $r[A_j]$ is the value of A_j for record r, $SW(A_j)$ is the sensitivity weight of A_j, and $SW(r[A_j])$ is the sensitivity weight of $r[A_j]$, the value of A_j in record r.

Equation (9) indicates that the consequence of a record r based on the i^{th} unknown set depends on sensitivity weights of the attributes in the i^{th} unknown set and the sensitivity weights of i^{th} unknown set's attribute values of the record r. Unknown sets comprising high sensitive attributes incur high consequence values for the records having high sensitive attribute values as well. Therefore, they have higher impact on FADR measure, compared to lower sensitive unknown sets. Hence, data publisher has the flexibility to make influence on FADR measure by assigning sensitivity weights to attributes and their values.

3.1.5 FADR Bound and Normalization

In this section, we want to measure the maximum and minimum risk of disclosure that can happen for a victim in any dataset with specific size and attributes, based on our FADR measure. Such bound is defined according to the specified publicly known probabilities and sensitivity weights for the attributes. Having such parameters set, FADR measure maximizes when the known tuple of the victim appears only once in a dataset and the corresponding unknown tuple disclosed has the maximum sensitivity weight of 1. Therefore, the maximum FADR is formulated as,

$$MaxFADR = \sum_{i} \left(PK(KS_i) \times 1 \times \sum_{A_j \in UKS_i} \left(SW(A_j) \times 1 \right) \right) \tag{10}$$

The minimum FADR value for a victim depends on the same parameters as well as the size of confidential dataset. FADR measure minimizes when the known tuple of the victim appears in all the records of the dataset and the corresponding unknown tuples disclosed have negligible sensitivity weights. Since such weights are between 0 and 1, we can consider 0.001 as a trivial value.

$$MinFADR = \sum_{i} (PK(KS_i) \times \frac{1}{size\ of\ dataset} \times \sum_{A_j \in UKS_i} (SW(A_j) \times 0.001)) \quad (11)$$

With FADR bound being specified with the assigned parameters, we can normalize the derived FADR values of the records in a dataset, as shown in Eq. (12), to be between 0 and 1.

$$NormFADR(r) = (FADR(r) - MinFADR)/(MaxFADR - MinFADR) \quad (12)$$

3.1.6 An Illustrative Example

In this section, we give an example of calculating likelihood and consequence of a known and unknown set of a record in the sample microdata, shown in Table 2. Since there are 5 attributes in the sample microdata, and each attribute can be appeared in the known set and unknown set, totally there are $2^5 - 2 = 30$ known and unknown sets of attributes. For example, {age, gender, race} can be a known set and {income, disease} would be the complement unknown set. Another known set can be {age, gender, race, income} with {disease} as the complement unknown set.

We assigned probabilities and sensitivity weights as shown in Table 3. In practice, the data publisher has the flexibility to set these values based on the underlying dataset. For instance, according to our assigned values in Table 3, we consider attribute *gender* to be 80% probable to be publicly known, whereas *disease* is set to be 0.1% probable to be publicly known for the members of this dataset. Therefore, a known set containing *gender* will lead to higher likelihood and thereby more contribution in risk measure, compared to a known set including *disease*. As another example, we set attribute sensitivity weight of 90 for *income* and 0 for *race*. This incurs a higher consequence value and thereby has more impact on FADR measure for an unknown set containing *income* than for one having *race.* In addition, records having income values less than 40K or more than 70K, are set to have higher value sensitivity weights than records in the middle-income levels, and thereby higher consequence value for an unknown set containing *income*. Value sensitivity weights for age, gender, and race are not defined in Table 3 because in computing consequence derived from Eq. (9), value sensitivity weights will be multiplied with attribute sensitivity weights that are 0 for these attributes.

Table 2. Sample microdata

	Age	Gender	Race	Income	Disease
r_1	34	Male	Black	60K	Flu
r_2	19	Female	White	36K	Flu
r_3	40	Male	Asian-Pac-Islander	45K	Flu
r_4	34	Male	Black	50K	Cancer
r_5	51	Female	Black	65K	Flu

Table 3. Probabilities and sensitivity weights for sample microdata

Attribute	Publicly Known Probability	Attribute Sensitivity Weight	Value Sensitivity Weight	
			Values	**Weight**
Age	0.3	0		-----
Gender	0.8	0		-----
Race	0.6	0		-----
Income	0.005	90	<40K or >70K	1
			[40K-70K]	0.7
Disease	0.001	100	Flu	0.2
			Cancer	1

Based on Eq. (10) and Eq. (11), the maximum and minimum FADR value for a victim's record incurred by any confidential dataset with the same size (n=5) and attributes (age, gender, race, income, and disease) as the sample microdata in Table 2, along with the assigned publicly known probabilities and attribute sensitivity weights shown in Table 3, are calculated below.

$$MaxFADR = \sum_{i=1}^{30} (PK(KS_i) \times \sum_{A_j \in UKS_i} SW(A_j)) = 178.87$$

$$MinFADR = \sum_{i=1}^{30} (PK(KS_i) \times \frac{1}{5} \times \sum_{A_j \in UKS_i} (SW(A_j) \times 0.001)) = 0.035$$

Likelihood and consequence for r_4 in the sample microdata, based on the known set {age, gender, race} and the unknown set {income, disease}, are derived as:

$$L_{\{age,gender,race\}}(r_4) = (0.3 \times 0.8 \times 0.6 \times (1 - 0.005) \times (1 - 0.001)) \times (1/2) = 0.071$$

$$C_{\{income,disease\}}(r_4) = (90 \times 0.7) + (100 \times 1) = 163$$

3.2 Calculation Efficiency

FADR measure, shown in Eq. (4), is calculated over all possible known/unknown sets of attributes – i.e., $2^m - 2$ number of sets where *m* is the number of attributes in the dataset. Calculating FADR measure can be computationally expensive in datasets with large number of attributes, due to the

exponential growth in the number of known/unknown sets. To make FADR measure computationally feasible, we reduce the number of known/unknown sets by pruning the branches of sets having very low contribution in the risk measure. We propose a pruning algorithm, which removes known/unknown sets of attributes, which have very low value of the product of likelihood and consequence.

As shown in Eq. (7), the second term of likelihood depends on the record values and value sensitivity weights in Eq. (9) make consequence values different for different records. Our pruning algorithm should prune the known/unknown sets, which result in very low disclosure risk contribution for any record. Therefore, we assume the worst record, which incurs the highest likelihood and consequence for any known/unknown set. We prune known/unknown sets incurring very low product of likelihood and consequence for the worst record and thereby the pruned known/unknown sets would have very low product of likelihood and consequence for any other records. The worst record has the maximum value of the second term of likelihood and sensitivity weight, which both are the value 1. Hence, the worst record disclosure risk measure will be simplified as

$$WR\_FADR_{KS_i/UKS_i} = PK(KS_i) \times \sum_{A_j \in UKS_i} SW(A_j), \tag{13}$$

and our pruning algorithm prunes branches of known/unknown sets having WR_FADR value of less than a threshold ε.

Since $PK(KS_i)$ is based on the product of probabilities, it follows that if one probability is less than a value, any probability multiplied with that will result in a smaller value. Therefore, we construct a tree of attributes, order by their publicly known and unknown probabilities. Each node is either a known or an unknown attribute and represents the subset of the i^{th} known and unknown set made from the node's attribute and all the attributes in the ancestors. The height of this tree is equal to the number of attributes (m). Therefore, each leaf node represents the complete i^{th} known/unknown set and $PK(KS_i)$ is derived at each leaf node by multiplying probabilities of the leaf and its ancestors. Traversing tree from the root to the leaves, attribute probabilities (PK and PUK) are monotonically decreasing. Our algorithm calculates the partial $PK(KS_i)$ at each node by multiplying the node's probability with the probabilities of the ancestors. If the partial $PK(KS_i)$ is less than a value, all the complete known/unknown sets represented in the leaves of the subtree at this node will have publicly known probabilities of a smaller value.

Our algorithm also needs to check the second term in (13), which represents the consequence. However, this term is not monotonically decreasing in the constructed tree. Therefore, at each node, by looking at the attributes in the descendants, we find the complete unknown set with the maximum consequence value, which is the sum of unknown attribute sensitivity weights. Accordingly, traversing from the root to the leaves of the tree, the maximum consequence value either remains the same or decreases. Thus, at each node, we multiply the partial $PK(KS_i)$ with the maximum consequence value, and if it gets lower than a threshold ε, all the complete

known/unknown sets represented in the leaves of the subtree at the node, will have the value for this product lower than ε as well. Hence, our algorithm prunes the subtree at a node having this product lower than ε.

Table 4. Sorted list of all probabilities for the sample microdata

$PUK(A_5)$	$PUK(A_4)$	$PK(A_2)$	$PUK(A_1)$	$PK(A_3)$	$PUK(A_3)$	$PK(A_1)$	$PUK(A_2)$	$PK(A_4)$	$PK(A_5)$
99.9%	99.5%	80%	70%	60%	40%	30%	20%	0.5%	0.1%

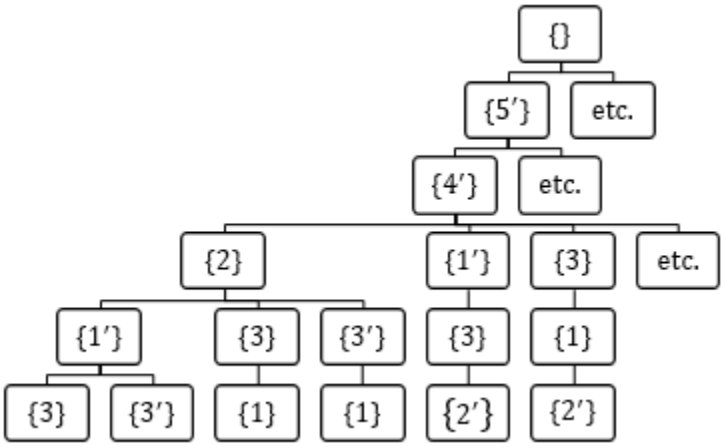


Figure 4. Part of the constructed tree for the sample microdata

For instance, back to the sample microdata in Table 2, and the assigned probabilities shown in Table 3, first we create a sorted list of all known and unknown probabilities, shown in Table 4. Then, we construct a tree accordingly. A part of a constructed tree is demonstrated in Figure 4 with attribute indices written in the nodes. In the tree, the attributes with the prime sign are classified into unknown set and the attributes without prime sign are classified into the known set. For example, the leftmost leaf node represents $\{A_2, A_3\}$ as the known set and $\{A_1, A_4, A_5\}$ as the unknown set.

Figure 5 shows an example of the way we prune a branch of the tree. This figure just shows a sample branch and does not show the subtree at the node. The publicly known probability of each subset is written next to each node, on the left column. The maximum consequence at each node is shown next to each node on the right column. At the node $\{5'\}$, a complete unknown set having the maximum consequence is the one with A_5 and A_4 as the unknown attributes, which results in 100+90 consequence value. However, at the node $\{3\}$, since the tree is in the order of probabilities, we don't have A_4 as the unknown attribute in the subtree and instead we have A_4 as the known

attribute. That is why the maximum consequence at this node is 100, which is just for having A_5 as the only unknown attribute. Consider the pruning threshold ε to be 2%. By moving towards the leaf and calculating the probabilities times maximum consequence, we can see that the probability of {3, 1, 4} known set and {5, 2} unknown set becomes 0.00017 and the maximum consequence value is 100. Therefore, 0.00017 × 100 equals to 0.017 which is less than 0.2 and therefore the tree is pruned from that node.

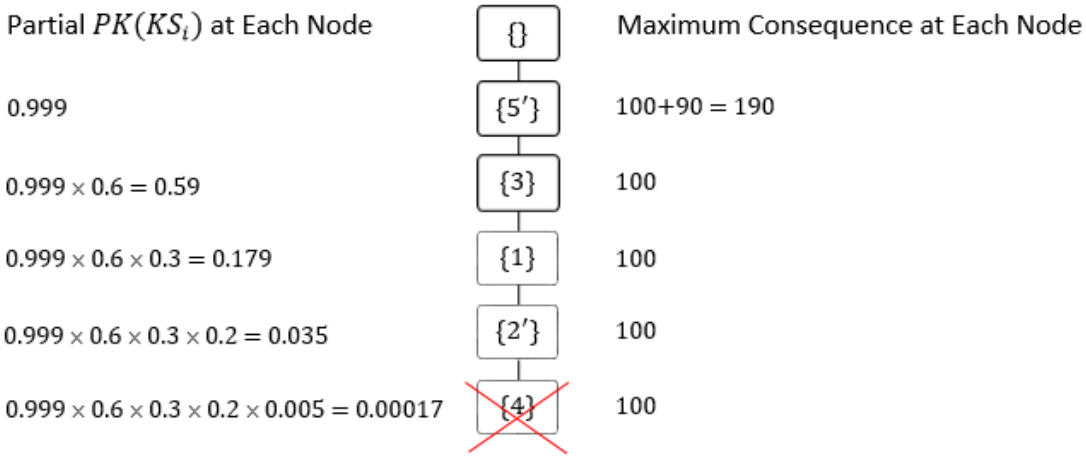


Figure 5. The pruning process example

Algorithm 1 shows our proposed pruning algorithm for removing known/unknown sets having low contribution in the risk measure. The inputs of our algorithm are the list of attributes in the dataset and the assigned publicly known probability and sensitivity weight for each attribute. The output is the list of known/unknown sets remained after pruning. Having publicly known probabilities of the attributes, we calculate publicly unknown probabilities based on (6). Then, we sort all the attributes, to both be known and unknown, in descending order of their probabilities, and write them in the variable S (line 2). For example, looking at the Table 4, S would be {5′, 4′, 2, 1′, 3, 3′, 1, 2′, 4, 5} (showing attribute indices only). Each attribute appears twice in this list, one to be known and one to be unknown (with prime sign). Therefore the length of S is $2m$. $SUBSETS$ contains subsets of known and unknown attributes with the WR_FADR value of (13) more than ε. $SUBSETS$ is initiated by an empty subset. $NEXTSETS$ contains subsets of attributes; each corresponds to one subset in $SUBSETS$, showing a set of remaining attributes that can be appended to the subset in $SUBSETS$ for enlarging the subset. $NEXTSETS$ is initiated by all attributes as the candidates to be appended to the empty set in $SUBSETS$, which is S. i is a pointer sweeping $SUBSETS$ and $NEXTSETS$ to point to the current subset and select it for checking whether it needs to be enlarged or pruned. i is initiated by 0 and line 5 of the algorithm shows that the first subset to select is the last set in $SUBSETS$. Pointer starts from the last set in $SUBSETS$ and sweeps

towards the first set. Whenever a new subset is appended to the $SUBSETS$, the pointer is reset to 0 to restart sweeping from the last set in $SUBSETS$.

After a subset is selected (line 5), its remaining attributes, selected from the corresponding set in $NEXTSETS$ (line 6), should be appended one by one to the selected subset and create a partial set to check (line 8). Each time the publicly known probability of the partial set (line 11) and the sum of sensitivity weights of the partial set and the next unknown attributes (line 14) are calculated, and the product of these two terms, based on (8), will be WR_FADR. We checked the value of WR_FADR (line 16). If it is more than ε:

- The partial set is appended to $SUBSETS$ (line 17).
- The set of remaining attributes for the partial set need to be appended to $NEXTSETS$ (line 18). The remaining attributes must not contain a known/an unknown attribute, which already exists as an unknown/a known in the partial set (line 12 and 13).
- If the length of the partial set is equal to the number of attributes, it implies that it is a complete known/unknown set and we have reached the leaf node of the tree. So, the algorithm outputs the set (line 19). Then, i is incremented by one (line 29) to select the prior subset in $SUBSETS$ (line 5) and its corresponding remaining attributes in $NEXTSETS$ (line 6) to process. If the length of the partial set is not equal to the number of attributes, we need to continue enlarging the partial set when there are still attributes remained for enlarging (line 7). Since the partial set was added to the end of $SUBSETS$, i is reset to 0 to start sweeping $SUBSETS$ and $NEXTSETS$ from the end.
  else:
- The partial set is not appended to the SUBSETS and not selected for enlarging. Therefore, the partial set is pruned and consequently all the known/unknown sets comprising this partial set are pruned.
- Then the algorithm continues with the subset selected earlier in line 5, and appends the next attribute to create a new partial set (line 8), if any attribute remained (line 7).

Whenever no more attributes remained for the selected subset (line 28), i is incremented by one (line 29) to select the prior subset in $SUBSETS$ and its corresponding remaining attributes in $NEXTSETS$ (line 5 and 6) and again these attributes are appended one by one and each time the WR_FADR is checked. This loop continues until no more subset is remained in $SUBSETS$ which its remaining attributes are not appended and their WR_FADR value are not checked (line 4).

Algorithm 1. Pruning Low Risk Sets

Input:

Attributes $\{A_1, A_2, \ldots, A_m\}$; Publicly Known Probability of Attributes $\{PK(A_1), PK(A_2), \ldots, PK(A_m)\}$; Sensitivity Weight of Attributes $\{SW(A_1), SW(A_2), \ldots, SW(A_m)\}$; Pruning threshold ε;

Output:

Remaining known/unknown sets after pruning;

Algorithm:

$\{PUK(A_1), PUK(A_2), \ldots, PUK(A_m)\} = \{1 - PK(A_1), 1 - PK(A_2), \ldots, 1 - PK(A_m)\}$;
S = sorted list of attributes to be both known and unknown in descending order of attribute's PK and PUK;
$SUBSETS = \{\{\}\}$; $NEXTSETS = \{\{S\}\}$; $i = 0$;
while $(length(SUBSETS) - i) != 0$ **do**
    $sub = SUBSETS(length(SUBSETS) - i)$;
    $nxt = NEXTSETS(length(NEXTSETS) - i)$;
    while $length(nxt) != 0$ **do**
        $PartialSet = sub$ union first attribute in nxt;
        Delete first attribute in nxt;
        $NEXTSETS(length(NEXTSETS) - i) = nxt$;
        $PK\_PartialSet = PK(PartialSet)$ calculated from (3);
        Remove known attributes in nxt which exist in $PartialSet$ as unknown;
        Remove unknown attributes in nxt which exist in $PartialSet$ as known;
        $MaxConseq$ = sum of sensitivity weights of unknown attributes in $PartialSet$ and nxt;
        $WR\_FADR = PK\_PartialSet \times MaxConseq$
        if $WR\_FADR > \varepsilon$ **then**
            Append $PartialSet$ to $SUBSETS$;
            Append nxt to $NEXTSETS$;
            if $length(PartialSet) == m$ **then**
                Output $PartialSet$
                Break;
            else
                $sub =$ last set in $SUBSETS$;
                $nxt =$ last set in $NEXTSETS$;
                $i = 0$;
            end if
        end if
    end while
    $i++$
end while

3.3 Experiments

In our experiments, we used a sample dataset from the UCI machine learning repository Adult dataset*, extracted from the 1994 Census database. Our Adult sample dataset contains 9 attributes of *Age*, *Work class*, *Education*, *Marital status*, *Occupation*, *Race*, *Gender*, *Country*, *Income*, and 30162 records after eliminating missing values.

Table 5. Assigned PKs and SWs for Adult sample dataset

Attribute	Publicly Known Probability	Sensitivity Weight of Attributes	Sensitivity Weight of Attribute Values	
			Values	**Weight**
Age (A_1)	0.3	0	all values	0
Work class (A_2)	0.1	100	without pay	1
			Other than "without pay"	0.1
Education (A_3)	0.1	100	primary school	1
			Other than "primary school"	0
Marital status (A_4)	0.01	0	all values	0
Occupation (A_5)	0.05	100	all values	1
Race (A_6)	0.6	0	all values	0
Gender (A_7)	0.8	0	all values	0
Country (A_8)	0.2	0	all values	0
Income (A_9)	0.001	100	all values	1

To calculate FADR measure for our Adult sample dataset, we assigned publicly known probabilities and sensitivity weights to the attributes as shown in Table 5. We assumed the gender attribute is the most probable attribute to be publicly known about individuals (80%), followed by race (60%), age (30%), and country of origin (20%). We considered the occupation and income of individuals as their most private information, regardless of the values (sensitivity weight of 100 for the attributes and 1 for all the values). Work class and education of the underlying individuals can be assumed as their sensitive information based on their values. For example, if their work class is assigned "without pay", that implies they have no income. In addition, the individuals

* https://archive.ics.uci.edu/ml/datasets/adult

underlying our sample dataset are all older than 16 years old. Therefore, if their education is stated as “primary school” it reveals that they have dropped out of school. As a result, we assigned sensitivity weight of 1 to “without pay” work class and “primary school” education.

The experiments are conducted on a machine with Intel(R) Core(TM) i7-3632QM CPU (2.20 GHz) and 8 GB RAM, programmed with R 3.3.2 software.

3.3.1 Evaluating FADR measure

In this experiment, we evaluate the FADR measure on the Adult sample dataset. We derived the FADR values for all the records based on Eq. (4). We used the parameter values shown in Table 5 with the pruning threshold (ε) of 0.01. The sum of all the records’ FADR value that represents the whole dataset’s disclosure risk (Total FADR) is derived as 72,917. Besides, we obtained the normalized FADR value for each record according to Eq. (12). Figure 6 shows the frequency of the records within each category of normalized FADR values.

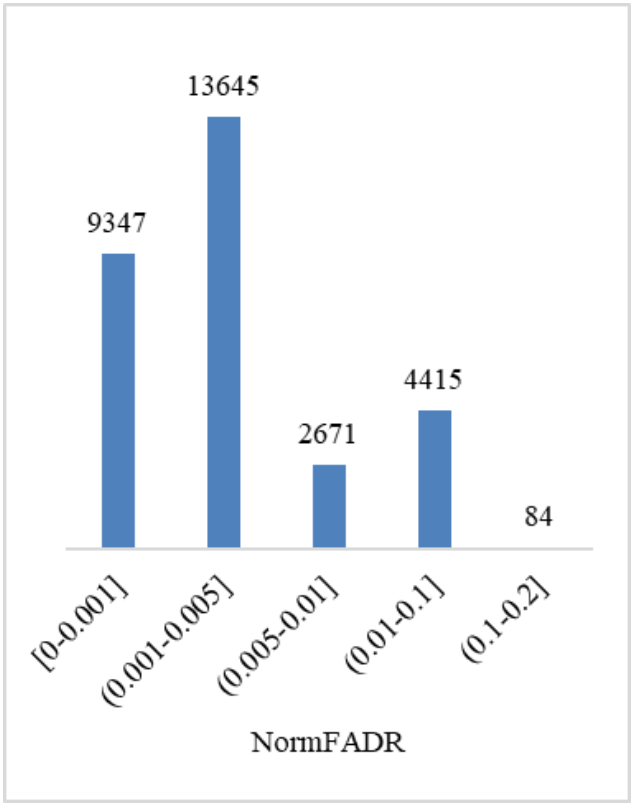


Figure 6. Distribution of the normalized FADR values among the records.

We identified a record $rMax$ in our dataset with the maximum FADR value, which is at the highest risk of disclosure. Such record has the normalized FADR value of 19.16%, and the FADR value

of 69.33. Looking at the values of $L_{KS}(rMax) \times C_{UKS}(rMax)$ for all the known/unknown sets, we found that the known set *{Age, Race, Gender}* and the corresponding unknown set *{Work class, Education, Marital status, Occupation, Country, Income}* has the highest value of 27.18 among all other sets, with the contribution of 39% ($27.18/69.33$) in the FADR calculation for this record. This implies the fact that 39% of the disclosure risk of $rMax$ is incurred by an adversary who knows the age, ethnicity, and gender of this victim without knowing the rest of information that are exposed to disclosure. The remaining 61% of FADR value is derived by considering other possible adversaries with different external knowledge and disclosure targets.

The likelihood value of $rMax$ obtained from the aforementioned known/unknown set is equal to 0.087, which means there is 8.7% of chance for the specified adversary to identify $rMax$ (identity disclosure). This value is derived from the probability of publicly knowing *{Age, Race, Gender}* and not knowing *{Work class, Education, Marital status, Occupation, Country, Income}* (equals 0.087) multiplied by the inverse frequency of the values of *{Age, Race, Gender}* for $rMax$ in our dataset (equals 1). $rMax$ has the known set values of *{63, other, male}*, which only appears once in the whole dataset. This is the first reason to make such record at the highest risk of disclosure.

Besides, the consequence value of $rMax$ incurred by the aforementioned unknown set, is equal to 310. This value indicates the level of disclosure that will happen if the victim of $rMax$ is found in the dataset (attribute disclosure). The unknown set of values for $rMax$ is *{Private, Preschool, Married-civ-spouse, Prof-specialty, Mexico, <=50}*, which contains the sensitive attributes of occupation (SW=100) and income (SW=100) regardless of the values, in addition to the sensitive attributes of work class (SW=100) and education (SW=100), for which the values are *private* (SW=0.1) and *preschool* (SW=1), respectively. Therefore, the consequence is derived as 100 + 100 + (100×0.1) + (100×1) = 310. The second reason that makes such record at the highest risk of disclosure is that all the sensitive attributes exist in the unknown set for the specified adversary and the record has the highest sensitive value for education attribute as well.

3.3.2 Likelihood Measure in FADR vs. Prosecutor Risk Measure

As described in Section 3.1.1, the likelihood term in FADR measure captures identity disclosure risk. Also, in Section 3.1.3, we mentioned that the prosecutor risk is a well-known identity disclosure risk measure and in Theorem 1 we proved that the prosecutor risk measure is a simplified version of the likelihood measure in FADR. However, the prosecutor risk measure restricts the adversary's knowledge to a specific set of attributes as QIDs. In Section 3.1.2, we addressed this limitation by showing that our approach considers all possible external knowledge for an adversary. The identity disclosure risk value measured by either prosecutor risk or our likelihood measure in FADR, is between zero and one.

In this experiment, we compared the likelihood measure in FADR, as a measure of identity disclosure risk, with the prosecutor risk measure, to evaluate the effect of considering all possible external knowledge for an adversary on the risk values of the records.

We calculated the prosecutor risk measure for each record in the Adult sample dataset by considering *{age, race, gender}* as the QIDs. This would result in the same likelihood measure in our approach if we assign PK of 1 to age, race, and gender attributes, and 0 to other attributes. In fact, it results in having only one set of known attributes which is *{age, race, gender}* and one set of unknown attributes comprising the remaining attributes. However, in our approach, we have the flexibility in assigning probabilities to consider any combination of attributes to be known and unknown. Thus, instead of assigning probabilities 1 and 0 to attributes, which restricts an adversary's external knowledge and disclosure target to only one known and unknown set, we changed PK of 1 to 0.99 and PK of 0 to 0.01, and calculated our likelihood measure in FADR for all the records.

Table 6. Comparing prosecutor risk with our likelihood measure.

Identity Disclosure Risk Measure	Number of known/unknown sets	Execution time (sec)	Average Identity Disclosure Risk Value
Prosecutor Risk	1	0.28	1%
Likelihood Measure in FADR	45	3.41	2%

Table 6 shows that changing the probabilities from 1 and 0 to 0.99 and 0.01 respectively, increased the number of known and unknown sets to 45 and the execution time is increased accordingly. Considering 45 known and unknown sets doubled the average identity disclosure risk. Therefore, generalizing the external knowledge and disclosure target of an adversary, incurs higher disclosure risk.

3.3.3 Evaluating our pruning algorithm

In this experiment, we evaluate the effect of pruning threshold (ε) on the total FADR, the number of known/unknown sets remained after pruning, and the execution time, shown in Table 7.

Figure 7 (a) shows that reducing the pruning threshold increases the number of remaining known/unknown sets because more sets will have the WR_FADR value of Eq. (13) larger than the small threshold. Since the number of sets increased, the execution time for calculating FADR measure will also be increased, as shown in Figure 7 (b), because there are more sets to count in the measure.

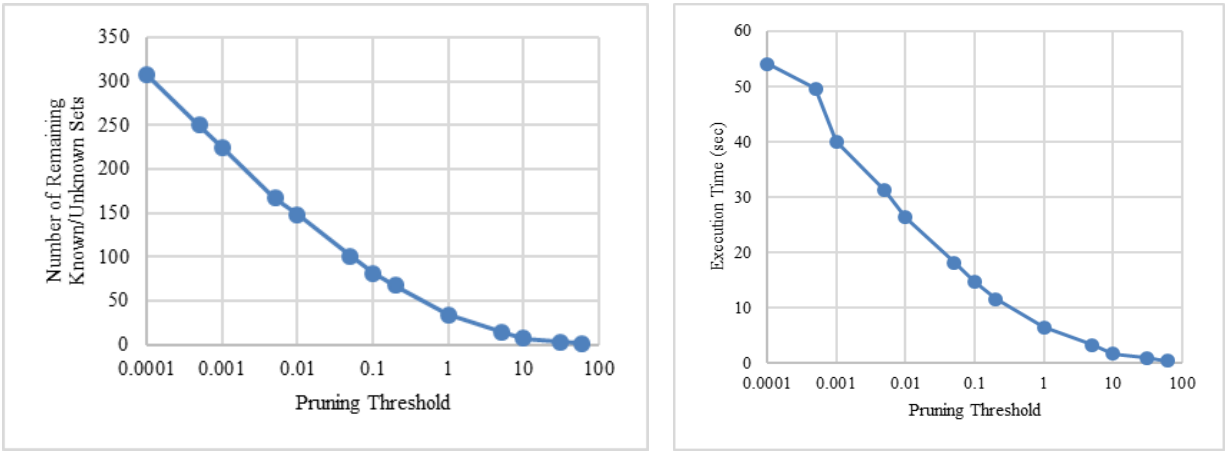
Table 7. Effect of pruning threshold on FADR, number of remaining sets, and execution time.

Pruning Threshold (ε)	Total FADR	Number of known/unknown sets	Execution time (sec)
0.0001	73,656.23	307	54.08
0.0005	73,620.73	250	49.67
0.001	73,569.47	225	40.09
0.005	73,255.3	167	31.28
0.01	72,917.68	148	26.36
0.05	71,240.02	101	18.17
0.1	69,358.27	81	14.69
0.2	66,350.64	67	11.55
0.999	52,968.05	34	6.53
5	32,639.34	14	3.25
10	14,769.63	7	1.75
30	10,419.69	3	1
60	436.43	1	0.43

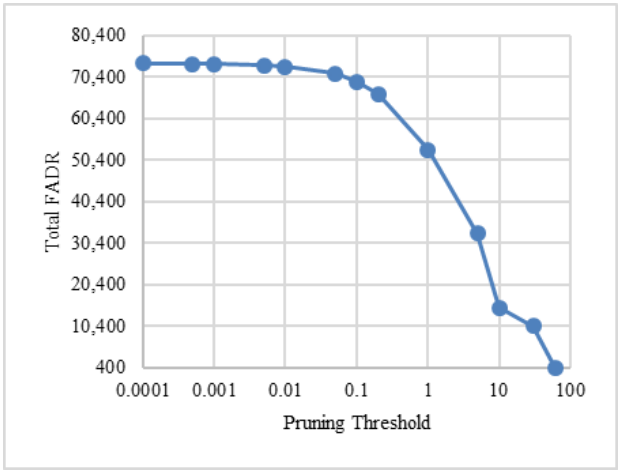
Figure 7 (c) shows that large thresholds result in lower total risk because we remove the known/unknown sets, which have high contribution in FADR measure. However, this figure shows that at some point ($\boldsymbol{\varepsilon}$ = 0.01 in this experiment), reducing the threshold does not change the total FADR. This proves that at this point, the pruning algorithm is removing the known/unknown sets with the low contribution in the risk measure. Besides, checking this point (ε = 0.01) in Figure 7 (a) and (b), we can see that the number of remaining known/unknown sets and execution time are still less than those of lower thresholds. Thus, it is concluded that the best pruning threshold to choose would be the one that reaches a total FADR value, which will no longer be increased by reducing the threshold whereas it results in lower number of remaining sets and execution time compared to the lower thresholds.

3.3.4 Effect of Publicly Known Probabilities

FADR is calculated based on the set of parameters assigned by the data publisher. One set of parameters is the publicly known probabilities. In this experiment, we first evaluate the effect of small changes in publicly known probabilities on the total FADR, and then illustrate how the total FADR changes if the data publisher assigns different values of probabilities.



(a) (b)



(c)

Figure 7. Effect of pruning threshold on (a) sum of records' FADR values (b) number of known/unknown sets remained, (c) execution time.

First, we apply small changes to each attribute's publicly known probability to evaluate the robustness of FADR measure to small changes in input parameters. Since this parameter is probability, we consider 1% as a small change, and increased each attribute's probability shown in Table 5 by 1%, one at a time, keeping the remaining probabilities and sensitivity weights constant. For each trial, we applied pruning algorithm with the same threshold of 0.01, and calculated FADR measure. The results are shown in Table 8. It is indicated that by increasing probabilities by 1%, the total FADR varies between 72,591.83 and 74,892.77. Figure 8 shows that the small changes of probabilities does not affect the total FADR value considerably. The highest change in total FADR occurs when the probability of attribute *Age* is increased because the age of

individuals in the dataset has more diversity (74 different ages) and if this information is known for a victim, fewer candidates can be matched with the same age for the victim, and therefore the risk of disclosure will be increased.

Table 8. Effect of small changes on publicly known probabilities.

Trials	Total FADR
No changes	72,917.68
$\boldsymbol{PK(Age)}$+0.01	74,892.77
$\boldsymbol{PK(Work\ class)}$+0.01	73,661.57
$\boldsymbol{PK(Education)}$+0.01	74,656.47
$\boldsymbol{PK(Marital\ status)}$+0.01	73,803.14
$\boldsymbol{PK(Occupation)}$+0.01	73,632.01
$\boldsymbol{PK(Race)}$+0.01	73,332.08
$\boldsymbol{PK(Gender)}$+0.01	73,191.58
$\boldsymbol{PK(Country)}$+0.01	74,186.3
$\boldsymbol{PK(Income)}$+0.01	72,591.83

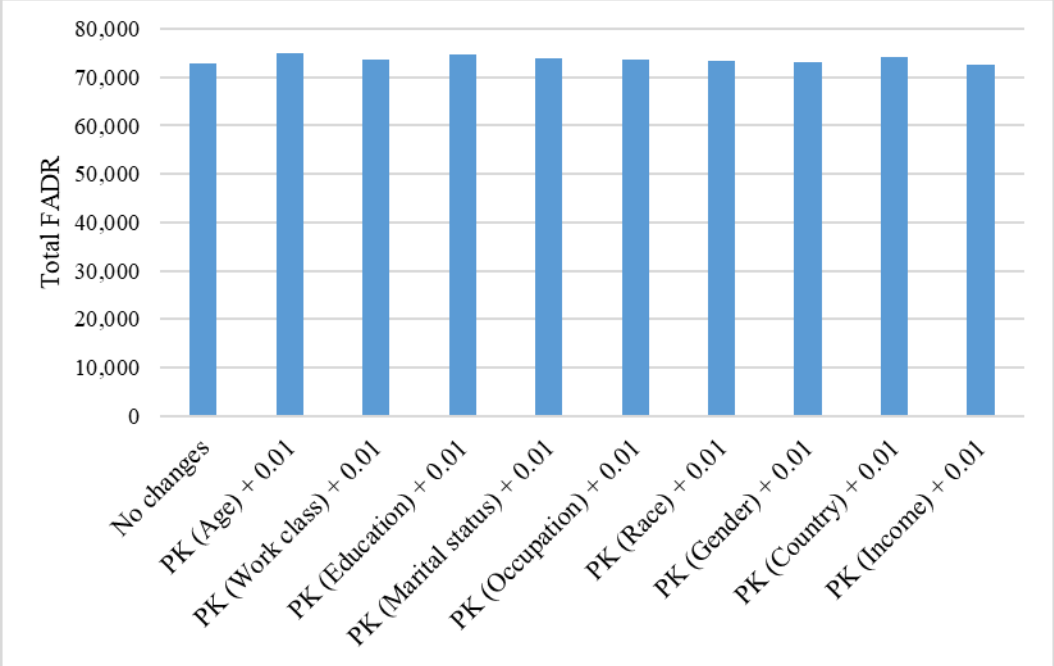


Figure 8. Comparing the total FADR values after applying small changes on probabilities.

In addition, we evaluated how the total FADR value of our dataset changes if the data publisher assigns different values of publicly known probabilities to attributes. For this experiment, we chose *Age* and *Income* attributes, as two representatives of insensitive and sensitive attributes, respectively.

The blue chart in Figure 9 shows that increasing publicly known probability of the *Age* attribute from zero to one, monotonically increases the total FADR, from 13,704 to 212,635. Comparing the trials, it is indicated that 10% increase in knowing the age of underlying individuals leads to the significant increase in the total FADR. Assigning higher probability to the *Age* attribute makes this attribute to be classified in the known set better than the unknown set after pruning. Since *Age* is not categorized as a sensitive attribute (SW=0), existing in the unknown set does not increase the consequence and thereby disclosure risk. Besides, since the underlying individuals have diverse age values, the frequency values of *Age* attribute become lower, which increases the likelihood of sets including *Age* as the known attribute. As a result, the more the *Age* attribute would be publicly known for the underlying individuals, the more disclosure risk occurs.

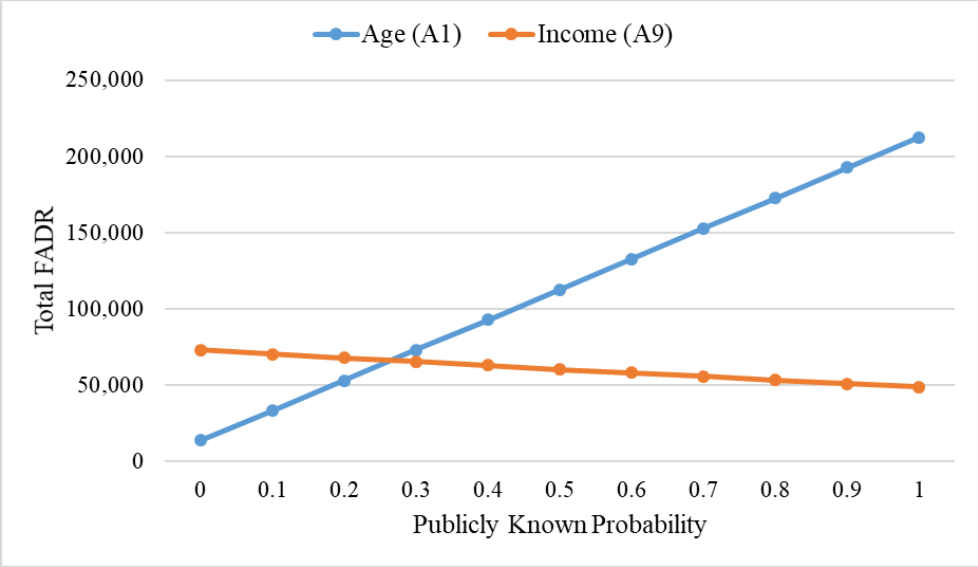


Figure 9. Comparing the total FADR values after assigning different publicly known probabilities to *Age* and *Income* attributes.

The orange chart in Figure 9 shows the same trials on the attribute *Income*. It shows the reverse trend as we increase the publicly known probability compared to the *Age* attribute. Increasing the probability from zero to one, monotonically decreases the total FADR from 72,977 to 48,788. Since the income level of individuals is set to be a sensitive information (SW=100), the more it appears in the unknown sets, the greater consequence occurs and the higher disclosure risk incurs.

By increasing the publicly known probability for this attribute, we classify it to more known sets than unknown sets, after pruning. Therefore, the total FADR is reduced.

3.3.5 Effect of Sensitivity Weights

In this experiment, first, we evaluate the robustness of FADR measure in small changes of the sensitivity weights. Then, we demonstrate how the total FADR value changes if the data publisher assigns different values of sensitivity weights.

As shown in Table 5, we initially assigned attribute sensitivity weight of 100 to *Work class*, *Education*, *Occupation*, and *Income* attributes, while the rest of attributes are classified as insensitive (SW=0). In this experiment, we applied small changes to both attribute and value sensitivity weights of the four sensitive attributes, one at a time, keeping the remaining parameters constant. Since we defined the attribute sensitivity weights must be between 0 and 100, small change of 1% requires the attribute sensitivity weights to be changed from 100 to 99 (1 unit reduction). In addition, we set the value sensitivity weights to be between 0 and 1. Therefore, as a small change of 1%, we subtracted 0.01 from the value sensitivity weights of the four attributes, except for the values other than "primary school" for *Education*, which we added 0.01 because their initial sensitivity weights were zero and we cannot assign value below zero as a weight.

Table 9. Effect of Small Changes on Sensitivity Weights.

Trials	**Total FADR**
No changes	72,917.68
$\boldsymbol{SW(Work\ class)}$-1	72,887.44
$\boldsymbol{SW(r[Work\ class]) - 0.01}$	72,617.62
$\boldsymbol{SW(Education)}$-1	72,903.95
$\boldsymbol{SW(r[Education] = primary\ school) - 0.01}$ $\boldsymbol{SW(r[Education] = all\ but\ primary\ school) + 0.01}$	73,147.03
$\boldsymbol{SW(Occupation)}$-1	72,605.54
$\boldsymbol{SW(r[Occupation]) - 0.01}$	72,605.54
$\boldsymbol{SW(Income)}$-1	72,544.6
$\boldsymbol{SW(r[Income]) - 0.01}$	72,544.6

The calculated total FADR values obtained in each trial of applying such small changes are shown in Table 9. It is illustrated that small changes on the sensitivity weights applied trivial changes on the total FADR value of our dataset (from 72,544 to 73,147.03), which proves the robustness of the FADR measure. We can see that the total FADR value remains the same when the sensitivity weight of *Occupation* reduced by 1 and the sensitivity weight of their values reduced by 0.01 (equals 72,605.54). The reason is that in the former case, the sensitivity weight for attribute and all the values are 99 and 1, respectively, and in the latter case, they are 100 and 0.99, respectively. In both trials, the product of such weights remains the same, and thereby it has the same contribution in FADR measure, which leads to the same total FADR values. The same reason applies to the two trails on *Income*.

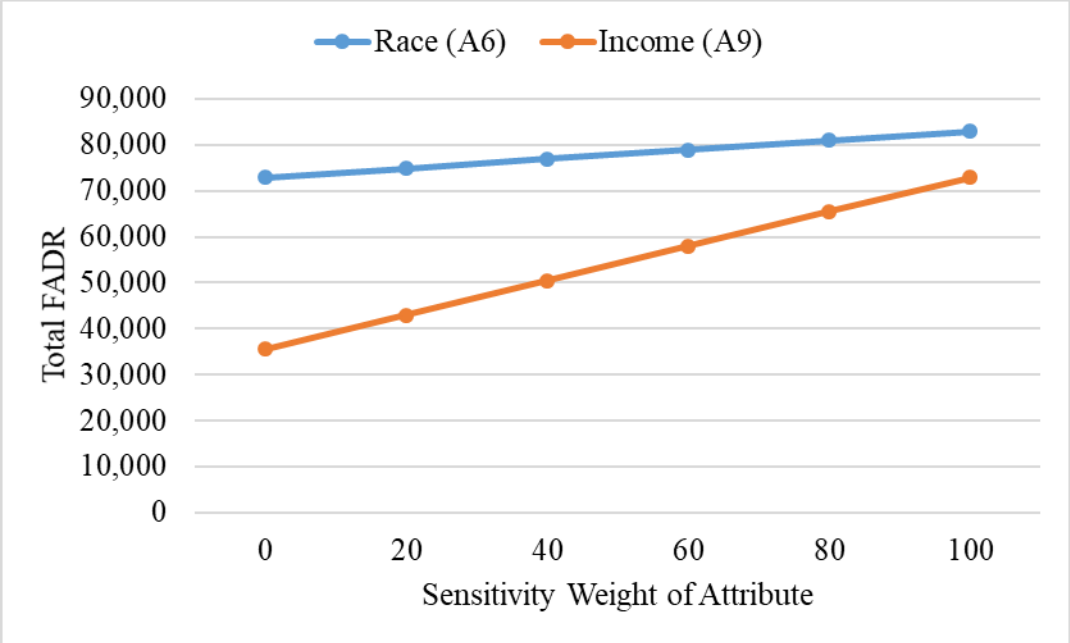


Figure 10. Comparing the total FADR values after assigning different sensitivity weights to *Race* and *Income* attributes.

In addition, we demonstrated the changes on the total FADR value when the data publisher assigns different values of sensitivity weights to attributes and attribute's values. Figure 12 shows the total FADR values achieved after assigning different values of attribute sensitivity weight to *Race* and *Income*. In both trials, we assigned value sensitivity weights of 1 to all races and income levels, and the rest of the parameters remain the same as shown in Table 5. It is illustrated that changing the sensitivity weight of *Income* attribute from 0 to 100, incus a larger increase in the total FADR (from 35,538 to 72,917) compared to the *Race* attribute (from 72,917 to 82,907). The reason is that according to Table 5, publicly known probability of *Income* attribute is much less than the *Race* attribute. Therefore, after pruning, *Income* appears more in the unknown sets than the known sets while *Race* exists in more known sets. Thus, *Income* has higher contribution in the consequence part of the FADR measure compared to the *Race*. As a result, sensitivity weight of

Income is more engaged in the FADR measurement than that of *Race*, which results in larger increase in total FADR when we change *Income*'s attribute sensitivity weight.

Figure 11 illustrates the changes in the total FADR, when data publisher changes the sensitivity weights assigned to attribute values of *Occupation* attribute. It is indicated that when we increase the sensitivity weight of all the values of *Occupation* from 0 to 1, the total FADR enhanced from 41,704 to 72,917.

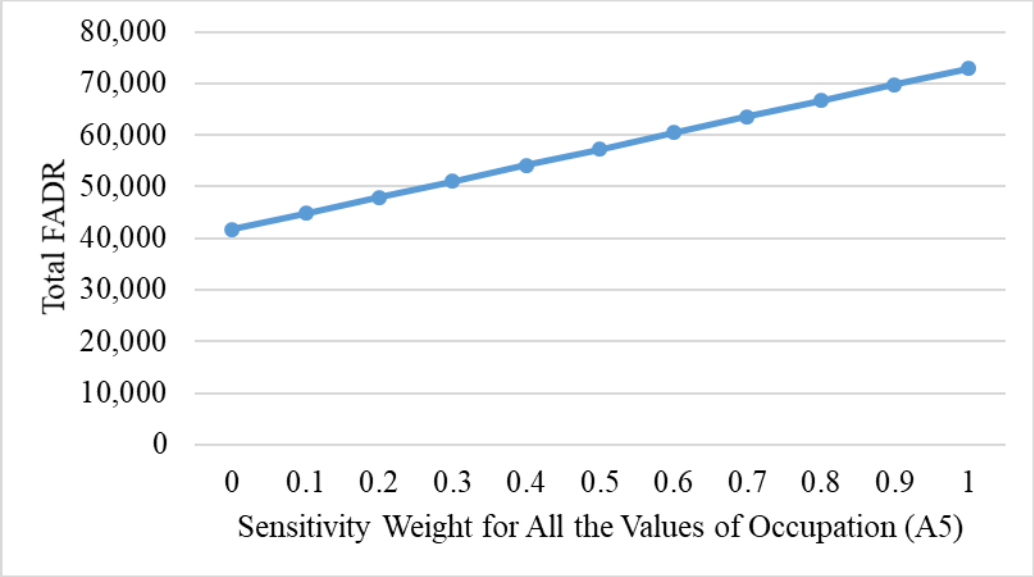


Figure 11. Comparing the total FADR values after assigning different sensitivity weights to all values of *Occupation* attribute.

3.4 Conclusions

In this chapter, we developed Flexible Adversary Disclosure Risk (FADR) measure, which captures both identity and attribute disclosure attacks concurrently. Besides, it models all possible external knowledge and disclosure targets for an adversary by considering any combination of attributes to be known and unknwon by an adversary. We proved that the likelihood term in FADR measure is a generalized form of the prosecutor risk for measuring the identity disclosure risk, which overcomes the limitation of the prosecutor risk measure in restricting adversary's external knowledge. Our approach also provides the flexibility to data publisher to assign publicly known probabilties and sensitivity weights to attributes to make known sets containing more probable attributes and unknown sets containnig more sensitive attributes have higher contribution in FADR calculation. Moreover, we developed a pruning algorithm to remove known/unknown sets having

very low contribution in the calcultaion of FADR measure, to reduce the computation complexity of our measure. Finally, through a set of experiments we showed the effectiveness of our pruning algorithm and the robustness of FADR measure to the small changes on the input parameters.

4 OPTIMIZING ANONYMIZATION

There is always a trade-off between preserving privacy and data utility, i.e., the more we anonymize the data to better preserve individual's privacy, the more information the data loses and the less data utility it preserves. As described in Chapter 2, various privacy techniques are developed in the literature to satisfy a privacy model by employing anonymization techniques through different algorithms based on data utility objectives. Such algorithms aim at both satisfying a privacy requirement and minimizing information loss. A privacy requirement is a Boolean condition that defines specific level of privacy preservation on a sensitive dataset.

In this chapter, we aim at developing an anonymization algorithm, which instead of achieving a predefined privacy level, minimizes the disclosure risk of a dataset, along with the information loss. Our anonymization algorithm, named "RU Generalization", is a risk/utility-based generalization algorithm that utilizes our developed FADR measure, described in Chapter 3, as a disclosure risk measure, and *Loss* measure [49, 50] (re-formulated as NCP [50]), as an information loss metric. We use local recoding generalization as the anonymization operation.

4.1 Data Utility Metric

In RU Generalization algorithm, we need a metric for measuring data utility to optimize anonymization with respect to preserving the most data utility as well as privacy. We have chosen *Loss* as the data utility metric to measure information loss on the generalized dataset. In the literature, *Loss* has been demonstrated as an effective measure of information loss on a generalized dataset, since it considers the generalization level of each value [39]. It defines information loss based on the magnitude of deviation of the generalized value from the original value, and penalizes more generalized values. *Loss* is defined at the attribute level, as shown in Eq. (14, and can be aggregated at the record (Eq.(15)) and dataset level (Eq. (16)).

$$Loss_A(r) = \begin{cases} \dfrac{range\ of\ r[A]}{range\ of\ A}, & A\ is\ numeric \\ \dfrac{leaf\ count\ at\ r[A]\ subtree\ of\ A's\ generalization\ \text{hierarchy}}{number\ of\ distinct\ values\ of\ A}, & A\ is\ categorical \end{cases} \quad (14)$$

Loss of an attribute's generalized value is calculated based on Eq. (14. If an attribute is numeric, *Loss* is derived by dividing the range of generalized value by the overall range of attribute's original values. For example, if the generalized value is [10-20] and the overall range of values is [0-50], the *Loss* of the generalized value equals 10/50. If an attribute is categorical, the generalization hierarchy tree of the attribute is considered, and the *Loss* is derived by dividing the number of leaves at the subtree of the generalized value in the generalization hierarchy tree by the number of distinct original values of the attribute. For instance, Figure 12 shows an example of

the generalization hierarchy tree for the attribute *Employment Type*. *Loss* of the generalized value *Government* equals 3/8 because the subtree at the node *Government* has 3 leaves and the attribute has 8 total distinct original values. *Loss* of an attribute is a value between 0 and 1. *Loss* of 1 for an attribute means the attribute value is suppressed.

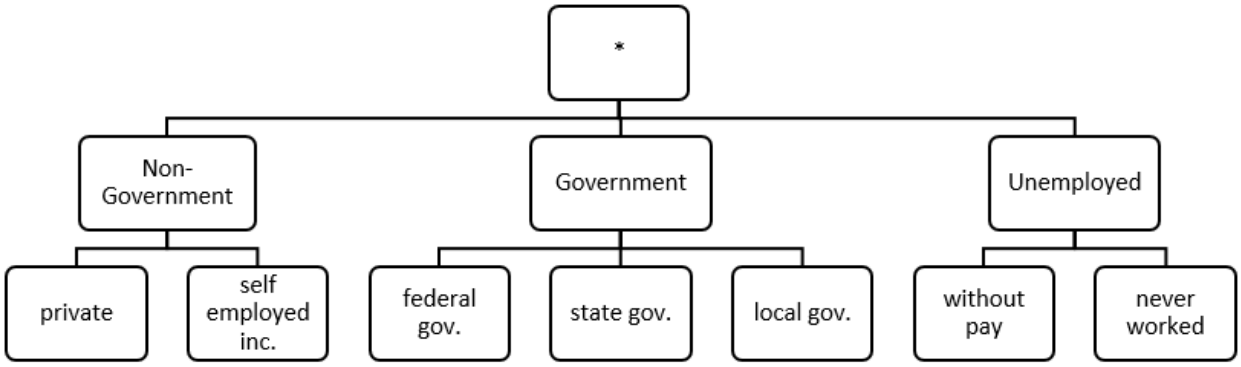


Figure 12. Generalization hierarchy tree for the attribute "Employment Type"

Once the *Loss* for the attributes are calculated, the *Loss* of a record is derived by the weighted sum of *Loss* of the record's attributes, as shown in Eq.(15), and is called the record's weighted *Loss*.

$$Loss(r) = \frac{\sum_i UW_i \times Loss_{A_i}(r)}{\sum_i UW_i} \tag{15}$$

Based on the domain area of the data use, different attributes can be at different levels of importance for the purpose of study. Some attributes might be of more importance for the use of users and therefore the users prefer to have the original values of such attributes. Therefore, the *Loss* of such attributes needs to be weighted over other attributes with lower importance, in aggregating the *Loss* for a record. The importance weight of each attribute is denoted as *UW* (Utility Weight) in Eq.(15).

$$Loss(D') = \sum_n Loss(r) \tag{16}$$

Finally, the *Loss* of the generalized dataset is the sum of all the records' weighted *Loss*, as shown in Eq. (16), and implies the total information loss of the generalized dataset D'.

4.2 RU Generalization Algorithm

Our developed RU Generalization algorithm is a greedy heuristic algorithm to obtain an optimum generalized dataset with the minimum disclosure risk and information loss, shown in Algorithm 2. At each iteration, our greedy algorithm targets a record with maximum disclosure risk and the corresponding set of attributes contributing the most in the risk measure. Then, it aims at generalizing such attributes by grouping with tuples that incurs the lowest information loss. After applying the generalization at each iteration, the FADR measure needs to be calculated to indicate the reduction in risk and identify the new record at the highest disclosure risk for the next iteration. FADR calculation requires the updated frequency counts on all the known sets of attributes. Since the data is generalized, such frequency counts are not merely capturing the exact matches. They also count the tuples that are within the generalized value. Therefore, recalculating the FADR measure at each iteration is computationally expensive. Thus, our heuristic algorithm estimates the FADR measure, at each iteration, by only updating frequency counts of the known sets contributing the most in FADR calculation of the records. Besides, the known sets that have tuples with high frequency counts are not updated, since increasing the large frequencies have small impact on the FADR reduction.

The inputs of RU Generalization algorithm include original dataset with n records (*D*), list of all known sets (*KS*), publicly known probabilities for all known/unknown sets (*PK*), likelihood and consequence of all records in the dataset for all known/unknown sets (*L* and *C*, respectively), FADR values for all the records on the original dataset (*R*), the list of generalization hierarchy trees for all categorical attributes in the dataset (*GH*), utility weights of attributes (*UW*), and the user defined maximum information loss (*MaxTotalLoss*). This algorithm outputs a generalized dataset that is optimum in terms of incurring the minimum summation of disclosure risk and information loss.

Our RU Generalization algorithm starts with initializing some variables (Algorithm 2, line 1-7). This algorithm applies generalization to the dataset at each iteration, and saves the generalized dataset as D', which is initially equal to the original dataset. $MaxTotalRisk$ is the total risk on the original dataset, which is the maximum total risk compared to the total risk of the generalized versions of the dataset, and is used to normalize total estimated risk. At each iteration, we estimate the records' disclosure risk. Therefore, we define the parameter $ERisk$ for saving the estimated FADR values for all the records, and initialize it to the original risk values. The total risk of the original dataset is the highest value compared to the total estimated risk values derived after generalizing the dataset. We normalize the total risk value to be between 0 and 1. Therefore, the initial total risk would be 1. The initial total *Loss* is 0, since we start with the original dataset with no information loss. DKS contains subsets of the original dataset for each set of known set attributes. FT is a matrix with rows for each record and columns for each known set and contains the frequency of known set tuples of records in the original dataset. Since our algorithm iteratively searches for the matches of known set tuples in the dataset, we create an index structure to store

the unique known set tuples for each known set as the search keys with the indices of matching records in the original dataset as the reference (Algorithm 2, line 5). This index table initially consists of the original tuples, and will be updated to contain generalized tuples with the indices for the matching original records at each iteration of generalization (Algorithm 5, line 22). The defined objective function is the summation of the data's total risk and total information loss (Algorithm 2, line 6).

Our optimization algorithm aims at minimizing our objective function. We iteratively generalize the dataset (Algorithm 2, line 8-29) and calculate the total estimated risk (Algorithm 2, line 17) and information loss (Algorithm 2, line 19) of the generalized dataset at each iteration and derive the objective function and append it to the previous ones (Algorithm 2, line 20).

At each iteration, we find the minimum of the objective function (Algorithm 2, line 21) and compare it to the global minimum obtained from the previous iterations (Algorithm 2, line 22). If the minimum value obtained in the current iteration is less than the global minimum from the previous iterations, it implies that the generalized dataset created in the current iteration has lower disclosure risk and information loss compared to the previous ones. Therefore, we save that as the optimum generalized dataset obtained so far (Algorithm 2, line 23), and update the global minimum point with the minimum value obtained in the current iteration (Algorithm 2, line 24).

Our algorithm continues until we know that the objective function will not go below the global minimum. As we further generalize the dataset, the total information loss always increases while the total risk decreases monotonically. Thus, if the total information loss gets larger than the global minimum of the objective function, it is concluded that, by further generalizing the dataset, our objective function will not go below the global minimum. As a result, our algorithm continues as long as the total information loss is less than or equal to the global minimum value of the objective function (Algorithm 2, line 8). When the total information loss gets larger than the global minimum value (Algorithm 2, line 26), the algorithm will not execute the next iteration and the generalized dataset which was obtained at the point of global minimum of the objective function is returned as the optimum generalized dataset (Algorithm 2, line 27).

At each iteration of our algorithm, we find a record in the dataset with the highest estimated risk value (Algorithm 2, line 9) and the corresponding known set of attributes that contributes the most in the risk calculation (Algorithm 2, line 10). A set of attribute values contributing the most in the risk calculation for the highest risk record is saved as *t1* (Algorithm 2, line 11). By scanning all sets of values for the same attributes in the dataset, we find tuples (save as *t2*) which incur low information loss if we group them with *t1* and generalize their attributes to have the same values (Algorithm 2, line 12). "Low Loss Tuples" is an algorithm we developed for finding such tuples, which is shown in Algorithm 4, and will be described in Section 4.2.1.

Algorithm 2. RU Generalization

Input: $D = \{r_1, r_2, \dots, r_n\}$, $KS = \{KS_1, KS_2, \dots, KS_m\}$, PK, L, C, R, GH, UW, $MaxTotalLoss$

Output: $OptimumGeneralizedData$

Algorithm:

1: $D' = D$; $MaxTotalRisk$ = sum(R); $ERisk = R$; $TotalERisk$ = 1; $TotalNCP$ = 0;
2: LC = matrix of $L \times C$ for all sets (each corresponds to one column) and all records (each corresponds to one row);
3: DKS = subsets of D for all known sets in KS;
4: FT = matrix of frequency of the known set tuples of all the records in D (number of rows = n, number of columns = number of known sets in KS);
5: $IndexStructure$ = for each known set, unique set of known set attribute values of D as the search keys along with the record's indices matching the tuple as the references;
6: $ObjFun = TotalERisk + TotalNCP$;
7: $GlobalMin = ObjFun$;
8: **while** ($TotalLoss \leq GlobalMin$) **do**
9: hr = record with maximum value in $ERisk$;
10: A = set of attributes in $KS[\underset{i}{\text{argmax}}\ (L[hr,] \times C[hr,])]$**;**
11: $t1 = hr[A]$ in D';
12: $t2$ = **Low Loss Tuples**($t1$, A, D, D', GH, UW);
13: GA = set of attributes having different values in $t1$ and $t2$;
14: gr = group of records in D' containing $t1$ and $t2$ for A;
15: $GENgr$ = **Generalize** (gr, GA, GH);
16: $IndexStructure$, FT, L, $ERisk$ = **FADR Estimate**(FT, GA, KS, DKS, gr, $GENgr$, GH, $IndexStructure$, PK, C);
17: $TotalERisk = sum(ERisk)$ / MaxTotal$Risk$;
18: D' = replace gr records in D' with $GENgr$;
19: $TotalLoss$ = sum of *Loss* of all attributes on the generalized records, calculated from Eq. (14)/$MaxTotalLoss$;
20: Append ($TotalERisk + TotalLoss$) to $ObjFun$;
21: $NewMin$ = min($ObjFun$);
22: **if** $NewMin < GlobalMin$ **then**
23: $OptimumGeneralizedData = D'$;
24: $GlobalMin = NewMin$;
25: **end if**
26: **if** $TotalLoss > GlobalMin$ **then**
27: Output $OptimumGeneralizedData$;
28: **end if**
29: **end while**

Among the attributes in *t1* and *t2*, the ones that have different values are going to be generalized and are saved as *GA* (Algorithm 2, line 13). Group of records in the dataset that contain *t1* and *t2* are saved as *gr* (Algorithm 2, line 14), and are generalized through "Generalize" algorithm. The generalized group is saved as *GENgr* (Algorithm 2, line 15). Algorithm 3 shows the generalization algorithm. It goes over the generalized attributes of the group. If the attribute is numeric, it generalizes the group attribute values to be the range of values in the group (Algorithm 3, line 3). For instance, if attribute values of the records in the group are 63,65, and 66, then the generalized value will be 63-66. It is possible that some records in the group have already been generalized from previous iterations. For example, the group records can be 62-64, 65, and 66. Then, the generalized value will be 62-66. If the attribute is categorical, we refer to the generalization hierarchy tree of the attribute, and the generalized value will be the lowest common ancestor of the values of the records in the group (Algorithm 3, line 5).

When we obtain the new generalized group (*GENgr*), we need to estimate the disclosure risk that incurs on the dataset with the new generalized group. The new FADR values for the original records based on the generalized dataset is estimated through "FADR Estimate" algorithm, shown in Algorithm 5, which will be explained in Section 4.2.2. Then, we calculate the total normalized FADR value of the generalized dataset, by dividing the summation of records' estimated FADR values by the maximum total risk, which is the total risk of the original dataset (Algorithm 2, line 17).

Algorithm 3. Generalize

Input: $Records, GenAttr, GenTree$
Output: $generalized\ records$
Algorithm:
1: **for each** $attr$ in $GenAttr$
2: **if** $attr$ is numeric **then**
3: $Records[, attr]$ = concatenate min($Records[, attr]$) and "-" and max($Records[, attr]$);
4: **else**
5: $Records[, attr]$ = lowest common ancestor of $Records[, attr]$ values in $GenTree$ associated with $attr;$
6: **end if**
7: **end for**
8: Output $Records$;

Afterwards, we update the records that have been selected for generalization with the generalized values in the dataset (Algorithm 2, line 18). Having the new generalized dataset, we calculate the total information loss, by adding the *Loss* of all the generalized attributes in D', based on Eq. (16). Similar to normalizing the total risk, we need to normalize the total information loss. However, the maximum information loss ($MaxTotalLoss$) is a parameter that can be defined by data

publisher (Algorithm 2, line 19). Therefore, it is possible that the total information loss of a generalized dataset exceeds the defined maximum total *Loss*, which leads to normalized total *Loss* value of more than one. Based on Eq.(14, the maximum *Loss* of an attribute is 1, which happens when the attribute value is suppressed. For instance, if the data publisher sets $MaxTotalLoss$ to be equal to *n* (number of records in the dataset), a generalized dataset with *n* suppressed values is considered to have maximum information loss.

Algorithm 4. Low Loss Tuples

Input: $t1$, A, D, D', GH, UW
Output: $t2$
Algorithm:
1: Initialize $AllLoss$ as an empty vector of size *n*;
2: **for each** $record$ in D'
3: $recordLoss$ = 0;
4: **for each** $attr$ in A
5: $tree$ = GH associated with $attr$;
6: **if** $attr$ is numeric **then**
7: $gMin$ = min($t1[attr], record[attr]$);
8: $gMax$ = max($t1[attr], record[attr]$);
9: $attrLoss$ = $(gMax - gMin)/range(D[, attr])$;
10: **else**
11: g = lowest common ancestor of $t1[attr]$ and $record[attr]$ in GH associated with $attr;$
12: $attrLoss$=leaf counts of subtree at g in $tree$/leaf counts in $tree$;
13: **end if**
14: $recordLoss$ = $recordLoss$+ $(UW[attr] \times attrLoss)$;
15: **end for**
16: $WeightedRecordLoss$ = $recordLoss/sum(UW[A])$
17: Append $WeightedRecordLoss$ to $AllLoss$;
18: **end for**
19: **if** min$(AllLoss) \leq 0.1$ **then**
20: $t2 = D'[(AllLoss \leq 0.1),]$;
21: **else if** min$(AllLoss) \leq 0.2$ **then**
22: $t2 = D'[(AllLoss \leq 0.2),]$;
23: ...
24: **else**
25: $t2 = D'[(AllLoss \leq 1),]$;
26: **end if**
27: Output $t2$;

4.2.1 Finding Tuples with Low Loss

At each iteration of the RU Generalization algorithm, we calculate the *Loss* measure of records in D' when they are grouped and generalized with *t1*. Then, the records which incur the lowest information loss after generalization are selected. This procedure is indicated in Algorithm 4. We first calculate the *Loss* of each attribute in *A,* based on Eq. (16). If the attribute is numeric, the minimum and maximum values of *t1* and D' attribute values are considered as the range in the numerator of the equation, and the range of attribute values in the original dataset (*D*) is counted in the denominator (Algorithm 4, line 7-9). If the attribute is categorical, the lowest common ancestor of *t1* and D' attribute values in the generalization hierarchy tree is considered as the generalized value, and the number of leaves at the subtree of the generalized value in counted in the numerator of the *Loss* formula (Algorithm 4, line 11-12).

Having the *Loss* of attributes in *A* calculated for all the records in D', we then calculate the weighted *Loss* for each record, by considering attributes' utility weights that are defined by the user, based on the Eq.(15) (Algorithm 4, line 16). We append the *Loss* of all the records in a vector (Algorithm 4, line 17), and records with low *Loss* (less than 0.1, 0.2, 0.3, etc. based on the values exist) are chosen as *t2* (Algorithm 4, line 19-26) to be grouped and generalized with *t1*.

4.2.2 FADR Estimate

At each iteration of RU Generalization algorithm, after creating a new generalized group, we need the new FADR values of the records and the total FADR of the dataset. We consider different adversaries who know different sets of attributes (known sets) about all the individuals in the original dataset. By generalizing some records, the number of candidates matching an original record increases. Therefore, the disclosure risk decreases. Back to the FADR measure defined in Chapter 3, the inverse frequency term in Eq. (7) decreases at each iteration of generalization.

In order to enhance the efficiency of our RU Generalization algorithm, we estimate the records' FADR value at each iteration, instead of calculating the exact values. Algorithm 5 shows the steps to estimate the FADR value of the records at each iteration of the RU Generalization algorithm. By considering the new generalized group at each iteration, the algorithm searches for the original records that are matched in the generalized group, and increase the frequency of such records. The estimation is considered by not updating the tuple frequencies of all the known sets. We only update the known sets contributing the most in the FADR calculation (Algorithm 5, line 3), as well as the known sets having tuple frequencies of low values. The latter means that the algorithm skips updating frequencies of the known sets with all tuple frequencies more than ε, because changes on the large frequencies have small impact on changing the FADR value (Algorithm 5, line 9-11). For FADR estimation, since we do not increase the frequency of some known sets, the total estimated FADR value is larger than the true value. Therefore, we over-estimate the total FADR value.

Algorithm 5. FADR Estimate

Input: $FT, GA, KS, LC, DKS, gr, GENgr, GH, IndexStructure, PK, C$

Output: Updated $IndexStructure$, Updated FT, Updated Likelihood, Estimated FADR

$OldFT = FT$;
$GenKS$ = known sets in KS containing any attributes of GA;
$MaxSets$ = known/unknown sets having max LC for records;
$TargetKS$ = intersection of $GenKS$ and $MaxSets$;
$D\_TargetKS = DKS[TargetKS]$;
$grSets$ = subsets of gr for all known sets in $TargetKS;$
$GENgrSets$ = subsets of $GENgr$ for all known sets in $TargetKS;$
for each i in $TargetKS$
    if all values in $OldFT[i] >$ ε **then**
        Next;
    end if
    $MatchRecords$ = empty vector;
    for each unique t in $GENgrSets[i]$
        $indx$ = 1:number of rows in $D\_TargetKS[i]$;
        for each $attr$ in t
            if $t[attr]$ is already generalized **then**
                $indx$ = **MatchGeneralizedGroup**($attr$, $t[attr]$, $D\_TargetKS[i][indx, attr]$, GH);
            else
                $indx$ = index of $D\_TargetKS[i][indx, attr]$ matching $t[attr]$;
            end if
        end for
        Append t in $IndexStructure[i]$ as a new search key with $indx$ as a reference;
        $indx$ = replicate $indx$ by the number of times t appears in $GENgrSets[i]$;
        $MatchRecords = MatchRecords$ union $indx$;
    end for
    $CountedRecords$ = references of tuples in $grSets[i]$ found in $IndexStructure[i]$;
    $NewMatchRecords$ = exclude $CountedRecords$ from $MatchRecords$;
    $MatchIndxFT$ = frequency table of $NewMatchRecords$;
    Add frequency values in $MatchIndxFT$ to $FT[, i]$ for the matched indices;
end for
Output updated $IndexStructure$; Output updated FT;
$L = PK \times \frac{1}{FT}$, for each record and each known set;
Output L;
$ERisk = \sum_{i=1}^{size(KS)} L_i \times C_i$;
Output $ERisk$;

In Algorithm 2, we initialized FT as the frequency table for all the records and all the known sets (Algorithm 2, line 4). In Algorithm 5, such frequencies are updated based on the new generalized group. The tuple frequencies of the known sets that contain the generalized attributes will only be updated ($GenKS$). In addition, to enhance the efficiency of our algorithm, we only update the frequency of known sets contributing the most in records' FADR values, i.e., having maximum value in LC matrix for the records ($MaxSets$). Therefore, the algorithm only updates the sets appear in both $GenKS$ and $MaxSets$, i.e., $TargetKS$ (Algorithm 5, line 4).

We also save the set of attribute values of $TargetKS$ in the original dataset as $D\_TargetKS$, and the records of the new generalized group, before and after generalization, as $grSets$ and $GENgrSets$, respectively (Algorithm 5, line 5-7). For each known set in $TargetKS$, we look at each unique tuple of $GENgrSets$, to find the original records in the corresponding $D\_TargetKS$ that match the generalized tuple. For attributes in the tuple that are not generalized, we just check which original records have the exact same values (Algorithm 5, line 19). For attributes in the tuple that are generalized, we need to see which original records have values that are embedded in the generalized value (Algorithm 5, line 17). This step is further explained in "Match Generalized Value" algorithm, shown in Algorithm 6.

Algorithm 6 indicates that if the checking attribute is numeric, it splits the generalized value into two numbers of minimum and maximum of the generalized range, and outputs the index of original records having numbers between the obtained minimum and maximum values (Algorithm 6, line 2-3). If the checking attribute is categorical, in the corresponding generalization hierarchy tree, it finds the leaf nodes at the subtree of the generalized value, and outputs the index of original records with attribute values that exist among the leaf nodes (Algorithm 6, line 5).

Algorithm 6. Match Generalized Value

Input:

$attr, GeneralizedValue, candidates, GH$

Output:

index of $candidates$ which are embedded in the $GeneralizedValue$

Algorithm:

1. **if** $candidates$ is numeric **then**
2. $MinMax$ = split $GeneralizedValue$ on "-";
3. $members$ = range of $MinMax$
4. **else**
5. $members$ = all the leaf nodes of the subtree at $GeneralizedValue$ in GH of $attr;$
6. **end if**
7. Output index of $candidates$ which are included in $members$;

Back to our "FADR Estimate" algorithm, once the matching original records found for a specific generalized tuple, the generalized tuple with the indices of the original matched records are appended to the $IndexStructure$ for the underlying known set (Algorithm 5, line 22). If the tuple appears more than once in $GENgrSets$, the indices of matching original records are duplicated by the number of times the tuple exists (Algorithm 5, line 23). Finally, after looking over all the tuples in $GENgrSets$, we find all the original records (with duplicates) that match the new generalized group (Algorithm 5, line 8-25). However, some of the found original matching records are the records that are already matched with the tuples before generalization. Therefore, such matches are already counted in FT from previous iterations. Thus, we need to exclude such record indices from the found pool (Algorithm 5, line 27). Such record indices are stored in the $IndexStructure$ from previous iterations, as the references to the tuples in $grSets$, which are the tuples before generalization (Algorithm 5, line 26).

Once we found all the records' indices that are newly matched in the new generalized group, we create a frequency table for such indices to see how many times each original record is matched in the new generalized group besides the previous matches (Algorithm 5, line 28). Then, we add such frequencies in the FT for the corresponding records and known sets (Algorithm 5, line 29). Since the FT and $IndexStructure$ are both updated in this algorithm, and will be used in the next iterations, this algorithm outputs the updated FT and $IndexStructure$ (Algorithm 5, line 31). After updating FT, we calculate and output the likelihood and FADR measure for all the original records (Algorithm 5, line 32-35).

Table 10. Sample adult dataset

	Age	Education	Employment Type
r_1	39	Bachelors	State-gov
r_2	25	HS-grad	Self-emp-not-inc
r_3	56	Bachelors	Local-gov
r_4	22	Some-college	State-gov
r_5	53	Bachelors	Self-emp-not-inc
r_6	49	HS-grad	Local-gov
r_7	67	HS-grad	Without-pay
r_8	24	1st-4th	Private
r_9	23	1st-4th	Private
r_{10}	66	5th-6th	Private

4.3 Illustrative Example

In this example, we illustrate the major steps of our RU Generalization algorithm. Assume a sample microdata, from Adult dataset, shown in Table 10, as our original dataset. At first, we calculate the FADR measure on the original dataset, based on assigned parameters shown in Table 11, and pruning threshold of 0.01. After applying pruning algorithm described in Chapter 1, 5 known/unknown sets are remained, as shown in Table 12.

The initial frequency table (*FT*) of all the records and all the 5 known sets is shown in Table 13. For instance, the second known set tuple of the first record is *{State-gov}* that appears 2 times in the dataset. The initial FADR values calculated for each record is shown in Table 14.

Table 11. FADR parameters for sample adult dataset

Attribute	**Publicly Known Probability**	**Attribute Sensitivity Weight**	**Value Sensitivity Weight**	
			Values	**Weight**
Age	0.3	0	all values	0
Employment Type	0.1	100	Without pay	1
			Other than "without pay"	0.1
Education	0.1	100	Primary school	1
			Other than "primary school"	0

Table 12. Known/unknown sets remained after pruning on the sample adult dataset

	Known Sets	**Unknown Sets**
1	{age}	{education, employment type }
2	{employment type}	{age, education}
3	{age, employment type }	{education}
4	{education}	{age, employment type }
5	{age, education}	{ employment type }

Table 13. Original frequency table of all known set tuples of all the records

	KS1	KS2	KS3	KS4	KS5
r_1	1	2	1	3	1
r_2	1	2	1	3	1
r_3	1	2	1	3	1
r_4	1	2	1	1	1
r_5	1	2	1	3	1
r_6	1	2	1	3	1
r_7	1	1	1	3	1
r_8	1	3	1	2	1
r_9	1	3	1	2	1
r_{10}	1	3	1	1	1

Table 14. Initial FADR values of the records on the original dataset

	age	education	type_employer	FADR
r1	39	Bachelors	State-gov	2.91
r2	25	HS-grad	Self-emp-not-inc	2.91
r3	56	Bachelors	Local-gov	2.91
r4	22	Some-college	State-gov	3.33
r5	53	Bachelors	Self-emp-not-inc	2.91
r6	49	HS-grad	Local-gov	2.91
r7	67	HS-grad	Without-pay	29.1
r8	24	1st-4th	Private	32.115
r9	23	1st-4th	Private	32.115
r10	66	5th-6th	Private	32.43

The RU Generalization algorithm, starts with the record with the highest FADR value, which is r_{10} and the known/unknown set contributing the most in *LC* calculation, which is the first known/unknown set that only contains the age attribute as the known attribute. Therefore, *t1* will be *{66}*, and after going through the "Low Loss Tuples" algorithm, r_7 is found to be merged with *t1* as the low loss group. Therefore, the *gr* is:

	Age	Education	Employment Type
r_7	67	HS-grad	Without-pay
r_{10}	66	5th-6th	Private

And after applying generalization, *GENgr* will be:

	Age	Education	Employment Type
r_7	66-67	HS-grad	Without-pay
r_{10}	66-67	5th-6th	Private

The "FADR Estimate" algorithm goes over all the known sets containing the generalized attribute (*Age* in this example), which are *KS1*, *KS3*, and *KS5*.

For *KS1* (*{age}*), the generalized group has two same tuples of *{66-67}*. Checking the *{age}* tuples in the original dataset, r_7 and r_{10} are found to be matched with this generalized tuple. Since *{66-67}* appears two times in the generalized group, we double the found matched record indices: *{*r_7, r_{10}, r_7, r_{10}*}*.

Table 15. IndexStructure of the first known set before generalization

Search Key	References
39	r_1
25	r_2
56	r_3
22	r_4
53	r_5
49	r_6
67	r_7
24	r_8
23	r_9
66	r_{10}

In addition, we need to search the tuples of *gr* in the *IndexStructure* to find the matching records that are already counted in *FT*. *IndexStructure* of the first known set (*{age}*) is shown in Table 15. *{66}* and *{67}* are the first known set tuples of *gr*. Searching in the *IndexStructure*, *{66}* has the reference of r_{10} and *{67}* has the reference of r_7. Therefore, *{*r_7, r_{10}*}* is excluded from *{*r_7, r_{10}, r_7, r_{10}*}*, and finally *{*r_7, r_{10}*}* is remained as the records that are newly affected by the generalization. Each of the affected records are found once, and therefore their frequencies in *FT* are added by 1. The updated *FT* and *IndexStructure* are shown in Table 16 and Table 17 , respectively.

Table 16. Updated frequency table after the first generalization on the first known set

	KS1	KS2	KS3	KS4	KS5
r_1	1	2	1	3	1
r_2	1	2	1	3	1
r_3	1	2	1	3	1
r_4	1	2	1	1	1
r_5	1	2	1	3	1
r_6	1	2	1	3	1
r_7	**1+1 = 2**	1	1	3	1
r_8	1	3	1	2	1
r_9	1	3	1	2	1
r_{10}	**1 +1 = 2**	3	1	1	1

Table 17. IndexStructure of the first known set after generalization

Search Key	References
39	r_1
25	r_2
56	r_3
22	r_4
53	r_5
49	r_6
67	r_7
24	r_8
23	r_9
66	r_{10}
66-67	r_7, r_{10}

4.4 Calculating FADR Exposed by an Anonymized Dataset

In the RU Generalization algorithm, we have over-estimated the FADR values at each iteration, by not updating all the known sets' frequencies. Once a generalized dataset is obtained from the RU generalization algorithm, the true FADR values need to be calculated for the records in the generalized dataset.

Our FADR calculation over an anonymized dataset follows the "maximum knowledge attacker model" [9], which considers an adversary who knows both the original and anonymized dataset and tries to do mapping between the two. Likewise, we start with the original dataset (D), and for each record and known set of attributes, we find matching records in the anonymized dataset. Once

we find frequency of the known set tuples of original records in the anonymized dataset, we can compute the FADR measure by multiplying the publicly known probabilities of the sets and the consequence values that are already calculated.

Algorithm 7. FADR Calculation

Input:

D, D', KS, PK, GH, C

Output:

FADR

Algorithm:

1: DKS = subsets of D for all known sets in KS;
2: $D'KS$ = subsets of D' for all known sets in KS;
3: FT = matrix of frequency of the known set tuples of all the records in D (number of rows = n, number of columns = number of known sets in KS);
4: **for each** i in KS
5: $MatchRecords$ = empty vector;
6: **for each** unique t in $D'KS[i]$
7: $indx = 1:n$;
8: **for each** $attr$ in t
9: **if** $t[attr]$ is already generalized **then**
10: $indx$ = **Match Generalized Value**($attr$, $t[attr]$, $DKS[i][indx, attr]$, GH);
11: **else**
12: $indx$ = index of $DKS[i][indx, attr]$ matching $t[attr]$;
13: **end if**
14: **end for**
15: $indx$ = replicate $indx$ by the number of times t appears in $D'KS[i]$;
16: $MatchRecords = MatchRecords$ union $indx$;
17: **end for**
18: $MatchIndxFT$ = frequency table of $MatchRecords$;
19: $FT[, i]$ = frequency values in $MatchIndxFT$ for the matched indices;
20: **end for**
21: $L = PK \times \frac{1}{FT}$, for each record and each known set;
22: $FADR = \sum_{i=1}^{size(KS)} L_i \times C_i$;
23: Output $FADR$;

Algorithm 7 indicates the steps to find the frequency of the known tuples of the original records in the generalized dataset. It is similar to Algorithm 5 (FADR Estimate), illustrated in Chapter 4, with the difference in the target generalized records. Since Algorithm 5 was called in each iteration of our RU Generalization algorithm, for the computation efficiency, we considered only the new generalized group to count the frequency of the matching original tuples, and we needed to exclude the matchings that were already counted in previous iterations. In Algorithm 7, we have the entire generalized dataset (D'), and for each generalized record and every known set, we find matching original tuples, through Algorithm 6 (Match Generalized Value). For each original record and specific known set, the frequency table is populated with the frequency value of the corresponding original tuple in the generalized dataset (line 19). PK and C, are publicly known probabilities of the known/unknown sets and consequence values of the records for each unknown set, respectively. The two parameters are pre-calculated based on the defined parameters for FADR measure, described in Chapter 3.

4.5 Experiments

In this experiment, we applied our RU Generalization algorithm on the same Adult sample dataset used in the experiment of Chapter 3. The input parameters of our algorithm in this experiment are shown in Table 18. The generalization hierarchy trees used for the categorical attributes are shown in Table 19-26.

Table 18. RU Generalization algorithm input parameter values

RU Generalization Algorithm Input Parameter	Value
$\boldsymbol{D}$	Adult sample dataset, 9 attributes and 30162 records (same dataset used in Chapter 3 experiment)
$\boldsymbol{KS, PK, L, C, R}$	Obtained from Chapter 3 experiment, with the FADR parameter values shown in Table 5 and pruning threshold of 0.01
$\boldsymbol{GH}$	Shown in Table 19-26
$\boldsymbol{UW}$	1 for all attributes
$\boldsymbol{MaxTotalLoss}$	$2n$, n, and $n/2$

In this experiment, since we are not considering specific use purpose of the dataset, we assume that all the attributes are in the same level of importance, and assigned the utility weight of 1 for all the attributes. For normalizing total *Loss*, we tested different values of $2n$, n, and $n/2$ for maximum total *Loss* parameter, to examine the output change of the algorithm. n is the size of our

Adult sample dataset, which equals 30,162. The total loss of n can be interpreted as a dataset with n suppressed values.

Table 19. Generalization Hierarchy of attribute "Race"

Level0	Level1
White	*
Asian-Pac-Islander	*
Amer-Indian-Eskimo	*
Other	*
Black	*

Table 20. Generalization Hierarchy of attribute "Work Class"

Level0	Level1	Level2
Private	Non-Government	*
Self-emp-not-inc	Non-Government	*
Self-emp-inc	Non-Government	*
Federal-gov	Government	*
Local-gov	Government	*
State-gov	Government	*
Without-pay	Unemployed	*
Never-worked	Unemployed	*

Table 21. Generalization Hierarchy of attribute "Gender"

Level0	Level1
Male	*
Female	*

Table 22. Generalization Hierarchy of attribute "Marital status"

Level0	Level1	Level2
Divorced	Spouse not present	*
Never-married	Spouse not present	*
Separated	Spouse not present	*
Widowed	Spouse not present	*
Married-spouse-absent	Spouse not present	*
Married-civ-spouse	Spouse present	*
Married-AF-spouse	Spouse present	*

Table 23 . Generalization Hierarchy of attribute "Education"

Level0	Level1	Level2	Level3
Bachelors	Undergraduate	Higher education	*
Some-college	Undergraduate	Higher education	*
Prof-school	Professional Education	Higher education	*
Assoc-acdm	Professional Education	Higher education	*
Assoc-voc	Professional Education	Higher education	*
Masters	Graduate	Higher education	*
Doctorate	Graduate	Higher education	*
1st-4th	Primary School	Primary education	*
5th-6th	Primary School	Primary education	*
Preschool	Primary School	Primary education	*
11th	High School	Secondary education	*
HS-grad	High School	Secondary education	*
9th	High School	Secondary education	*
7th-8th	High School	Secondary education	*
12th	High School	Secondary education	*
10th	High School	Secondary education	*

Table 24. Generalization Hierarchy of attribute "Occupation"

Level0	Level1	Level2
Sales	Nontechnical	*
Exec-managerial	Nontechnical	*
Handlers-cleaners	Nontechnical	*
Other-service	Other	*
Adm-clerical	Other	*
Farming-fishing	Other	*
Transport-moving	Other	*
Priv-house-serv	Other	*
Protective-serv	Other	*
Armed-Forces	Other	*
Tech-support	Technical	*
Craft-repair	Technical	*
Prof-specialty	Technical	*
Machine-op-inspct	Technical	*

Table 25. Generalization Hierarchy of attribute "Income"

Level0	Level1
<=50K	*
>50K	*

In this experiment, we ran our RU Generalization algorithm three times, with three different values of maximum total *Loss*. Figure 13-15 show the reduction in normalized total estimated FADR and the increase in normalized total *Loss*, as our algorithm iteratively generalizes the dataset, for the three trials. The optimum point chosen by the algorithm is cross marked in the figures, which is the point of global minimum of the objective function. As described in Algorithm 2, line 26, when the normalized total *Loss* gets larger than the global minimum, the algorithm terminates. This is indicated in Figure 13-15, where the last point of iteration has normalized total *Loss* value larger than the marked global minimum.

Table 26. Generalization Hierarchy of attribute "Country"

Level0	Level1	Level2
South	Africa	*
Cambodia	Asia	*
India	Asia	*
Japan	Asia	*
China	Asia	*
Iran	Asia	*
Philippines	Asia	*
Vietnam	Asia	*
Laos	Asia	*
Taiwan	Asia	*
Thailand	Asia	*
Hong	Asia	*
England	Europe	*
Germany	Europe	*
Greece	Europe	*
Italy	Europe	*
Poland	Europe	*
Portugal	Europe	*
Ireland	Europe	*
France	Europe	*
Hungary	Europe	*
Scotland	Europe	*
Yugoslavia	Europe	*
Holland-Netherlands	Europe	*
United-States	North America	*
Puerto-Rico	North America	*
Canada	North America	*
Outlying-US	North America	*
Cuba	North America	*
Honduras	North America	*
Jamaica	North America	*
Mexico	North America	*
Dominican-Republic	North America	*
Haiti	North America	*
Guatemala	North America	*
El-Salvador	North America	*
Ecuador	South America	*
Columbia	South America	*
Nicaragua	South America	*
Trinidad &Tobago	South America	*
Peru	South America	*

The only difference between the three trials is the value of maximum total *Loss*, which only affects the normalized values of the total *Loss* at each iteration. In all the three trials, the normalized total estimated FADR obtained at each iteration is the same. Therefore, the blue chart line in Figure 13-15 are all the same. The total *Loss* at each iteration for all the three trials are all the same as well. However, since the maximum total *Loss* is different, the normalized total *Loss* at each iteration is different in the three trials. Consequently, the objective function, which is the sum of normalized total FADR and normalized total *Loss*, has different trend in the three trials.

Smaller value of the maximum total *Loss* makes the value of total loss to be normalized to a larger value. Thus, as shown in Figure 13-15, when the maximum total *Loss* decreases from $2n$ to $n/2$, at each iteration the value of the normalized total *Loss* increases. Therefore, from Figure 13 to Figure 15, the slop of the orange chart line increases. As a result, since the blue chart line is constant in the figures, the minimum point of the grey chart line, which is the sum of blue and orange charts, happens at the earlier iterations, when the maximum total *Loss* decreases from $2n$ to $n/2$. In addition, larger values of the normalized total *Loss*, with the constant normalized FADR, lead to higher values of the objective function. Thus, from Figure 13 to Figure 15, the global minimum of the objective function increases.

For the trial of the $2n$ maximum total *Loss*, Figure 13 indicates that our algorithm stopped after 518 iterations, since the normalized total *Loss* (44.38%) went above the minimum point of the objective function (31.91%). The optimum generalized dataset is achieved at the 471st iteration, when the normalized estimated total disclosure risk is 22.74% (77.26% reduction) and normalized total infor**m**ation loss is 9.17% (9.17% increase).

For the trial of the n maximum total *Loss*, Figure 14 indicates that our algorithm stopped after 512 iterations, since the normalized total *Loss* (51.32%) went above the minimum point of the objective function (40.09%). The optimum generalized dataset is achieved at the 441st iteration, when the normalized estimated total disclosure risk is 26.27% (73.73% reduction) and normalized total information loss is 13.83% (13.83% increase). For this trial, after we obtained the optimum generalized dataset, at the 441st iteration, we ran the Algorithm 7, to find out the true total FADR value. The actual normalized total FADR value of the generalized dataset derived as 18.34%, which is 7.93% lower than the estimated value.

For the trial of the $n/2$ maximum total *Loss*, Figure 15 indicates that our algorithm stopped after 512 iterations, since the normalized total *Loss* (102.64%) went above the minimum point of the objective function (50.32%). As mentioned earlier, the normalized total *Loss* can exceed 100% when the total loss of a generalized dataset exceeds the user defined maximum total *Loss*. The optimum generalized dataset is achieved at the 353rd iteration, when the normalized estimated total disclosure risk is 34.10% (65.9% reduction) and normalized total information loss is 16.22% (16.22% increase).

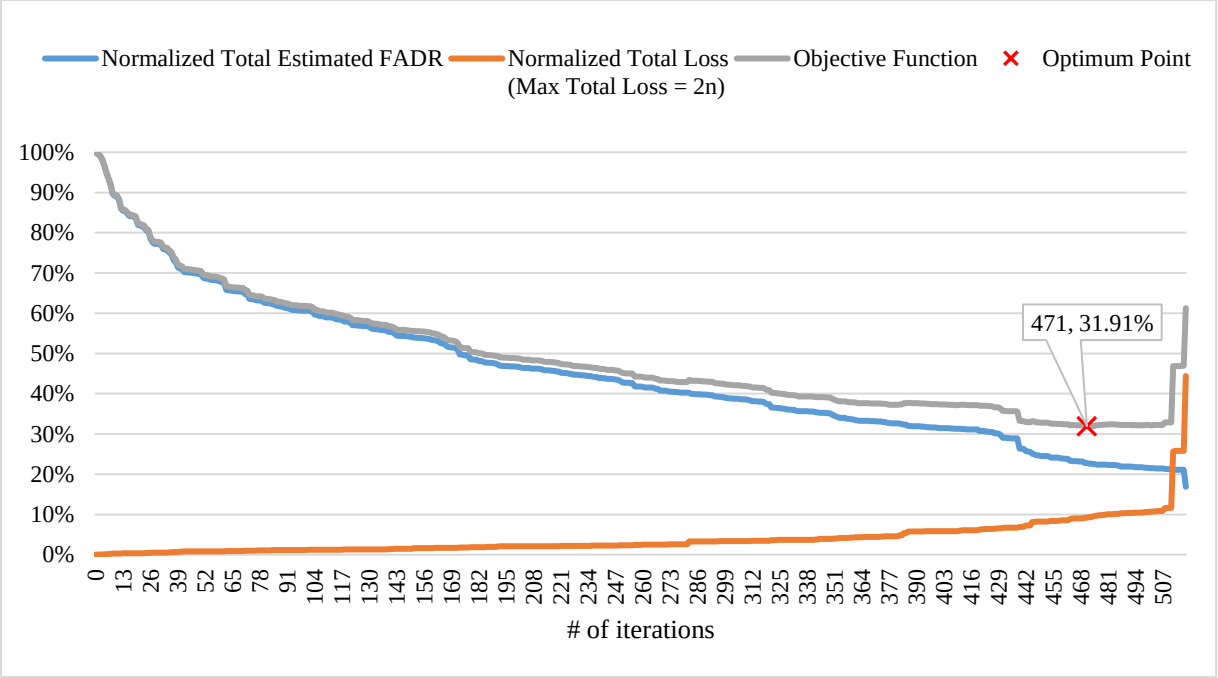


Figure 13. Trend of risk and information loss over the iterations of RU Generalization algorithm, when MaxTotalLoss = 2n

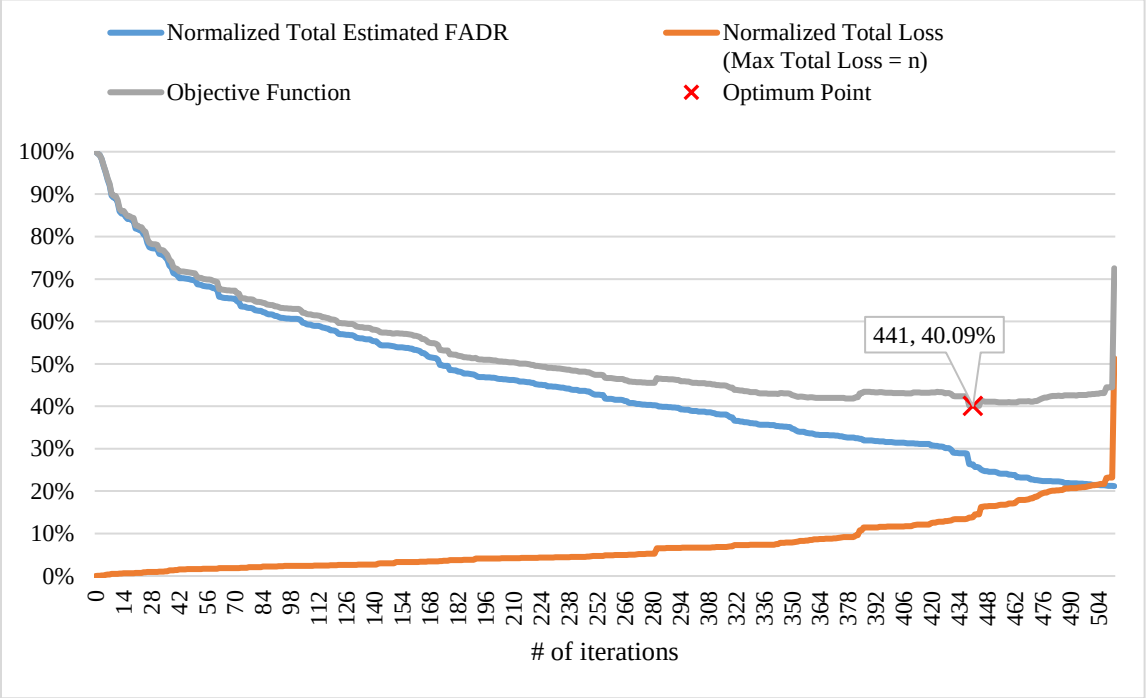


Figure 14. Trend of risk and information loss over the iterations of RU Generalization algorithm, when MaxTotalLoss = n

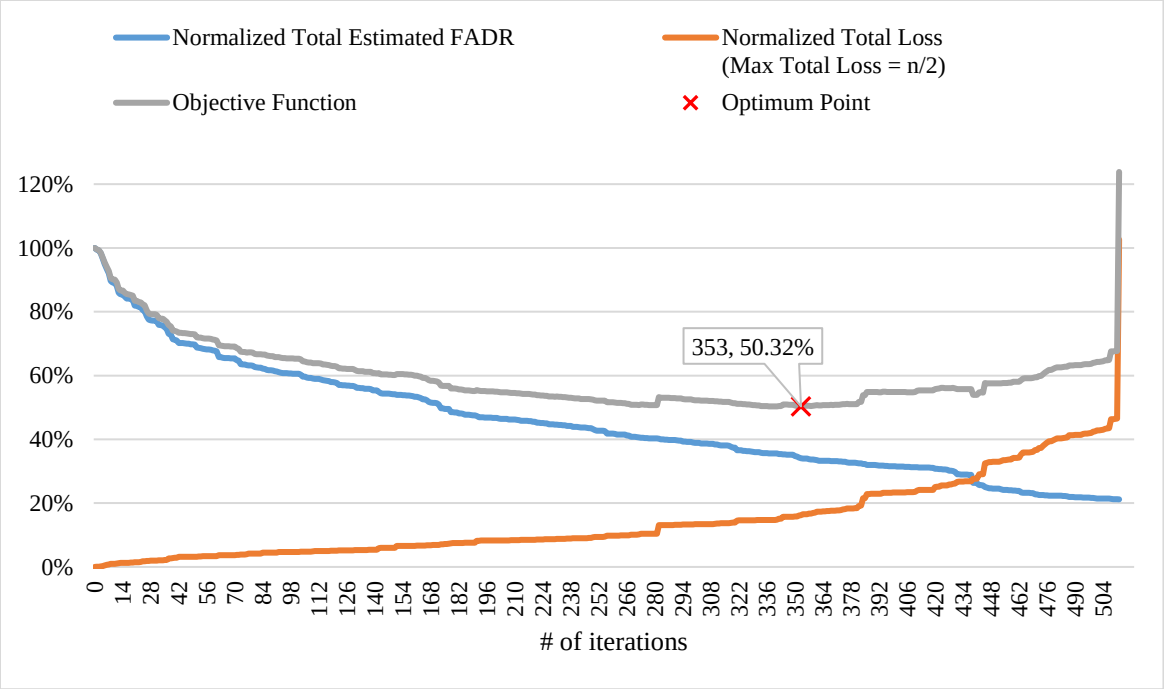


Figure 15. Trend of risk and information loss over the iterations of RU Generalization algorithm, when MaxTotalLoss = n/2

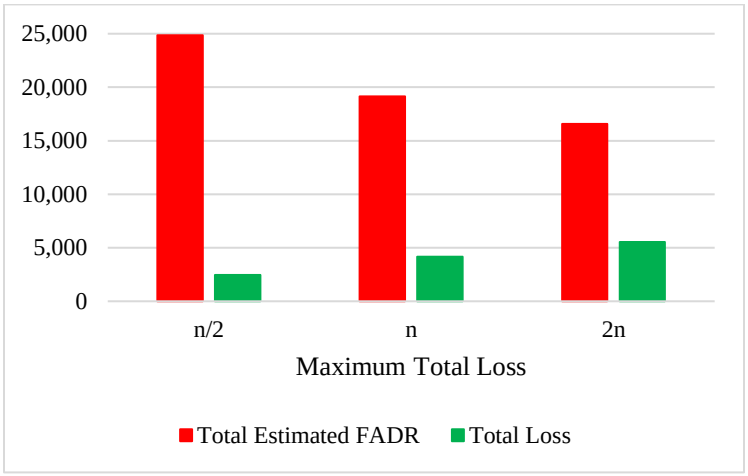


Figure 16. Effect of maximum total *Loss* on the non-normalized values of total estimated FADR and total *Loss* at the optimum point

Figure 16 shows the non-normalized values of the total estimated FADR and total *Loss* at the optimum point, in the three trials. Increasing maximum total *Loss*, makes the total loss values to be normalized to smaller values. Therefore, the information loss portion of the objective function

gets lower values at each iteration. As a result, our algorithm, with the higher maximum total loss assignment, iterates more to reach a lower objective function. Iterating more applies more generalization, which results in lower disclosure risk but higher information loss.

5 ANONYMIZED DATASET EVALUATION

An anonymized dataset is evaluated in terms of the amount of both privacy and data utility it preserves. Disclosure risk measure can be used to quantify privacy preservation, and information loss metric is a measure of data utility. In this chapter, we evaluate the generalized dataset obtained from our RU Generalization algorithm, described in Chapter 4, by calculating FADR and Loss measures, to evaluate privacy and data utility preservation, respectively. The total *Loss* of a generalized dataset is derived by the sum of Loss of all the attributes on the generalized records, calculated from Eq. (14, based on the defined generalization hierarchies for the attributes. The FADR measure of a generalized dataset is calculated through Algorithm 7, described in Chapter 4.

In this chapter, we also aim at comparing our RU Generalization algorithm with the benchmark generalization algorithms. We use the ARX Data Anonymization Tool [51], for implementing the benchmark local recoding generalization on Adult dataset, with average re-identification risk privacy model, to obtain different generalized datasets. We then compare the generalized datasets obtained from ARX Anonymization Tool with the generalized dataset obtained from our RU Generalization algorithm, with respect to the FADR and Loss measures.

5.1 ARX Data Anonymization Tool

ARX is a comprehensive software for anonymizing structured microdata based on user-defined privacy criteria, utility measure, and data transformation technique [52]. To be able to compare the results from ARX and our developed RU Generalization algorithm, we selected average re-identification risk measure for the privacy criteria, Loss for the utility measure, and local generalization as the transformation technique. Like our RU Generalization algorithm, generalization hierarchies for the attributes need to be defined in ARX to perform generalization and calculate data utility measures. ARX requires the users to specify the type of attributes, whether they are direct-identifier, quasi-identifier, sensitive, or insensitive attributes. This is required for benchmark anonymization algorithms, as described in the literature. However, in Chapter 3, we explained that classifying attributes is not practical since an adversary with any background knowledge and disclosure target may exist, and by classifying attributes we only model a specific adversary. We have addressed this issue in FADR measure by considering various known and unknown sets of attributes. Since our developed RU Generalization algorithm utilizes the FADR measure, different known and unknown sets are taken into consideration for anonymization in contrast to the ARX anonymization that only considers one known set of quasi-identifiers. Figure 17 shows an example screenshot of the ARX software for configuring the transformation. On the left, the input data is illustrated. On the right, the user needs to specify different parameters. For instance, the *age* attribute is classified as a quasi-identifier, and the transformation is selected to be generalization. The generalization hierarchy for the *age* attribute

is also imported on the right-hand side. Below the generalization hierarchy is where to add the privacy criteria and utility measures.

ARX employs the Flash algorithm to perform anonymization [53]. The Flash algorithm initially builds a generalization lattice by combining the generalization hierarchies. For example, consider the three attributes of *Employment type*, *Gender*, and *Age*. Example generalization hierarchies for such attributes are shown in Figure 18. It shows that *Employment type* has 2, *Gender* has 1, and *Age* has 5 levels of generalization. Based on such hierarchies, a generalization lattice is built as shown in Figure 19. Each node represents one transformation. For instance, (1,0,2) means the *employment type* is generalized to level 1, *gender* is not generalized (level 0), and *age* is generalized to level 2. Each level shown in the generalization lattice corresponds to the total level of generalizations of the transformations at that level.

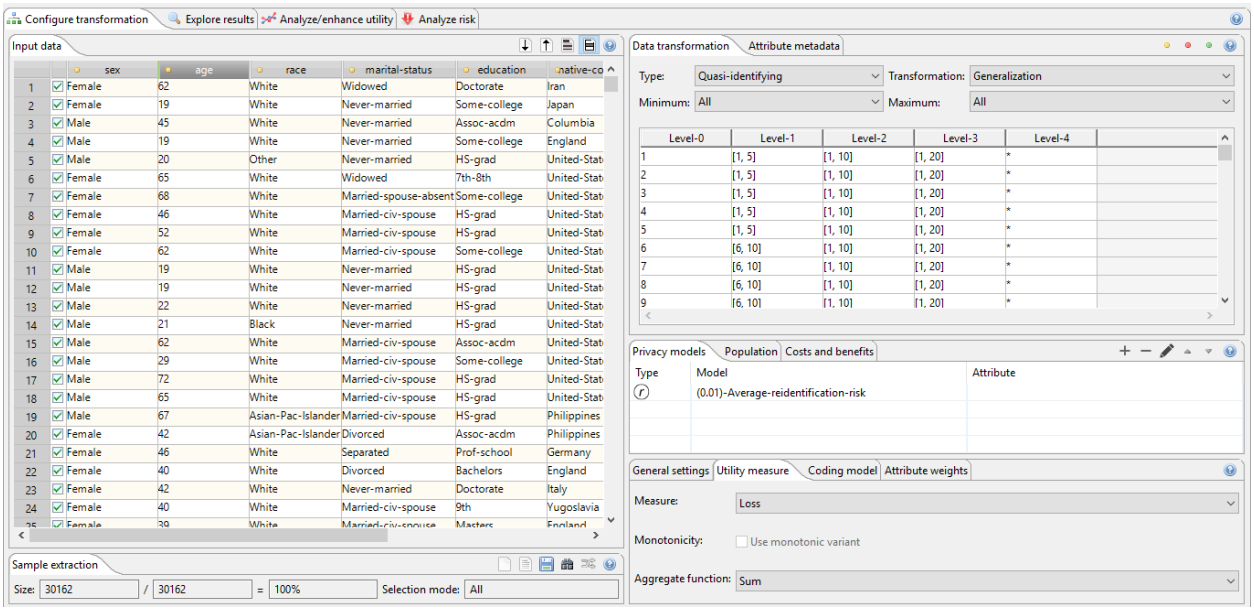


Figure 17. ARX transformation configuration

All the privacy criteria implemented in ARX have the monotonicity property, meaning that if a transformation meets the privacy criteria, all the successor' transformations in the generalization lattice meet the criteria as well. Likewise, if it does not meet the privacy criteria, all the predecessor's transformations do not meet the criteria. Therefore, whenever the Flash algorithm checks a transformation, based on whether it meets the criteria or not, it tags the predecessors or successors as either the candidate solutions or pruned transformations. For instance, in Figure 19, the dark gray nodes are tagged as pruned transformations while the light gray nodes are tagged as the candidate solutions. The tagged transformations will no longer be checked by the algorithm, which improves the efficiency of the algorithm.

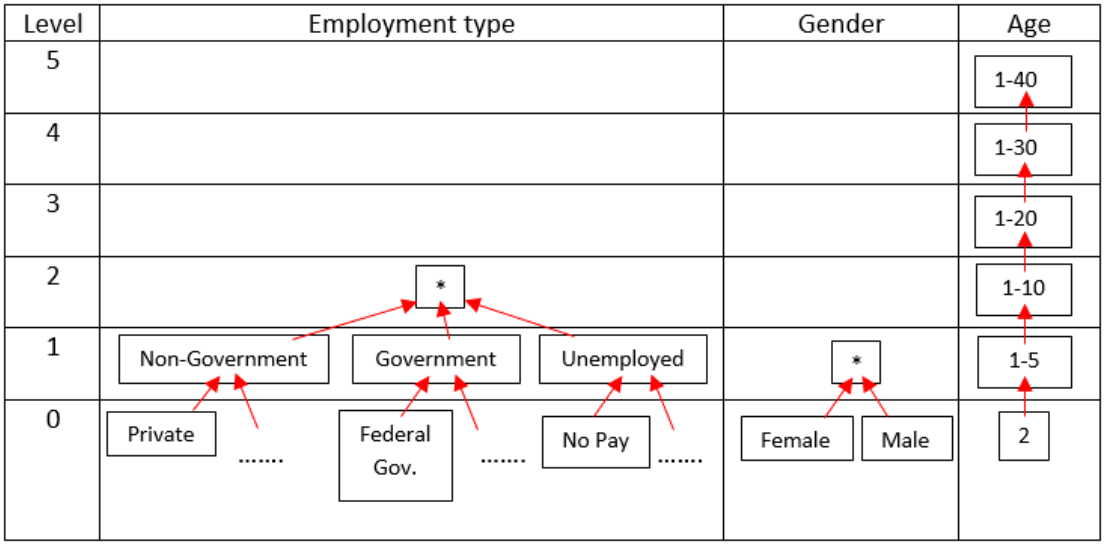


Figure 18. Example generalization hierarchies for building the generalization lattice

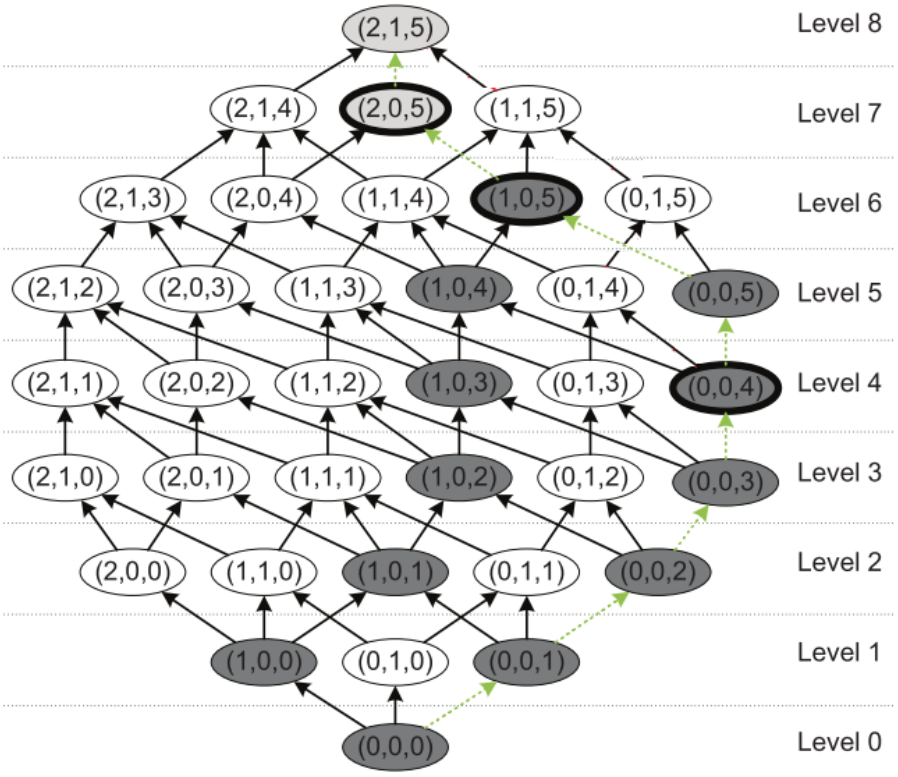


Figure 19. Illustration of the first iteration of Flash algorithm on the generalization lattice [53]

The Flash algorithm navigates through all the levels of the generalization lattice, from the lowest level to the top. At each iteration, it takes one node of the level, and creates a path of non-tagged nodes through a depth-first search towards the top node. Once the path is created, a binary search is implemented on the path. The search starts at the node in the middle level of the path and checks the privacy criteria. If the transformation meets the criteria, all the successors in the lattice are tagged as candidate solutions and the binary search goes into the lower half of the path. If the transformation does not meet the criteria, all the predecessors in the lattice are pruned and the binary search goes into the upper half of the path. The algorithm stops when all nodes are tagged. Among the candidate solutions, the one incurring minimum information loss is selected as the optimum solution.

Figure 19 shows the first iteration of the flash algorithm. It starts from (0,0,0) and creates a path to (2,1,5) illustrated with the dashed green flashes. For binary search, the first node to check the privacy criteria is (0,0,4). Since it does not meet the criteria, all the predecessors are pruned as shown in dark gray. Then, the node at the middle level of the upper half of the path is checked, which is (1,0,5). Again, it does not meet the criteria, and all the predecessors are pruned. The next node to check is (2,0,5), which meets the criteria, and therefore is tagged as a candidate solution along with all its successors (shown in light grey).

As mentioned earlier, we have selected average re-identification risk as our privacy criteria in ARX. This privacy model ensures that the average re-identification risk of records in the dataset, after anonymization, is less than a user-defined threshold. This criteria is checked in the following steps [52]:

1. Apply transformation to data.
2. While risk estimate is greater than the threshold:
    2.1. Suppress all the records in the equivalence class which incurs the highest information loss.
    2.2. Recalculate re-identification risk.
3. If the number of suppressed records is lower than a user-defined suppression limit, the privacy criteria is met and the transformation is a candidate solution.

The Flash algorithm applies global recoding generalization, which transforms all the values of a quasi-identifier to the same level of generalization. However, in ARX, we can enable the local recording feature to be added to the Flash algorithm. ARX will perform local recoding by recursively executing a global transformation algorithm on records that have been suppressed in the previous iteration. With this method, a significant improvement in data quality can be achieved.

5.2 RU Generalization Algorithm vs. ARX Average Re-identification Risk Model

The global transformation algorithm in ARX anonymization tool tends to incur higher information loss than our developed RU Generalization algorithm. In ARX, the generalization is applied to all the records, at each iteration, whereas our RU Generalization algorithm applies generalization to only a group of records with the lowest information loss that contains the highest risk record, at each iteration.

Our RU Generalization algorithm targets the records with the highest disclosure risk, at each iteration of generalization, to reduce the maximum disclosure risk by incurring low information loss. In ARX, the generalization is applied to all the records to reduce the total disclosure risk, without prioritizing the records at high risk of disclosure.

The average re-identification risk model in ARX only considers identity disclosure risk, whereas our RU Generalization algorithm works with FADR measure which is a combined measure of identity and attribute disclosure risk. In addition, ARX requires a pre-defined set of quasi-identifiers and sensitive attributes, which restricts the background knowledge and disclosure target of an adversary. FADR measure used in RU Generalization algorithm gives the flexibility in modeling adversaries by assigning different publicly known probabilities and sensitivity weights to attributes.

5.2.1 Experiments

In this section, we evaluate the disclosure risk and information loss of the generalized datasets obtained from our RU Generalization algorithm and ARX anonymization tool, and compare our algorithm with the benchmark algorithm implemented in ARX.

Table 27 shows the parameters we used in this experiment, for our algorithm and in ARX. We have executed the average re-identification risk privacy model in ARX, with Flash algorithm, on our Adult sample dataset, to obtain the local generalized dataset. As described earlier, in ARX, we need to define the type of attributes, and for average re-identification risk privacy model, specifying quasi-identifiers is required. In our FADR measure and RU Generalization algorithm, we do not specify the quasi-identifiers. Instead, we assign publicly known probabilities to attributes and build various known sets of attributes, with different probabilities, and consider all in FADR measure and RU Generalization algorithm. In order to make the setting of ARX and our RU Generalization algorithm similar, we choose the attributes that appear in the longest known sets to be quasi-identifiers in ARX. Such attributes are *{Age, Work class, Education, Occupation, Race, Gender, Country}.*

Table 27. Parameters of our RU Generalization algorithm and ARX anonymization tool

Algorithm	Parameters	
RU Generalization Algorithm	All parameters shown in Table 18, Except MaxTotalLoss = n	
ARX Anonymization tool	Dataset:	Adult sample dataset, 9 attributes and 30162 records (same dataset used in Chapter 3 experiment)
	Quasi-identifiers:	{Age, Work class, Education, Occupation, Race, Gender, Country}
	Privacy Model:	Average Re-identification Risk
	Maximum Average Re-identification Risk:	{0.01, 0.1, 0.2, 0.3}
	Information Loss Metric:	Loss
	Suppression Limit:	100%
	Local Recoding:	Enabled

It is also recommended, in ARX, to set the suppression limit to 100%, in order to balance the generalization and suppression to achieve the optimal solution [52].

As described earlier, the average re-identification risk privacy model in ARX ensures that the average re-identification risk of records does not exceed a maximum value predefined by user. In our experiments, we set the parameter to the values 0.01, 0.1, 0.2, and 0.3, to obtain different generalized datasets. Then, we calculated the FADR values of the obtained datasets, through Algorithm 7. For the calculation of likelihood and consequence, we used the same parameters of Chapter 3, shown in Table 5. The total FADR of the generalized dataset is then calculated as the sum of all the records' FADR values. We then normalized the total FADR, by dividing the values by the maximum total FADR, which is for the original dataset. In addition, we calculated the total *Loss* of the generalized datasets by adding the *Loss* of all attributes of the generalized records, calculated from Eq. (14, based on the attributes' generalization hierarchies, shown in Chapter 4. We considered the maximum information loss to be equal to *n*, which implies that the highest information loss occurs when we have *n* suppressed values in the generalized dataset. With such maximum *Loss* value, we normalized the total *Loss* values.

Table 28 illustrates the comparison of disclosure risk and information loss between the optimum generalized dataset obtained from our RU Generalization algorithm and ARX anonymization tool. The superscripts of ARX in the table represent the maximum average re-identification risk specified at each trial. The total FADR and total *Loss* measures in the table are normalized. Total FADR is normalized by being divided by the maximum total FADR, which is the total FADR of the original dataset. Total *Loss* is normalized by being divided by a user-defined maximum total *Loss* value. In the experiment, we set it to the size of dataset, which is *n*.

Table 28. Comparing FADR and *Loss* of the generalized datasets obtained from ARX and our RU Generalization algorithm.

	Max Record's FADR Value	**Normalized Total FADR %**	**Normalized Total Loss % (max total loss = n)**	**Objective Function (Total FADR % + Total Loss %)**	**Execution Time (sec)**
Original Data	69.16	100.00%	0.00%	100.00%	-------
$\boldsymbol{ARX^{0.01}}$	12.81	2.95%	178.65%	181.60%	3.08
$\boldsymbol{ARX^{0.1}}$	6.14	4.62%	69.22%	73.84%	2.42
$\boldsymbol{ARX^{0.2}}$	25.98	7.43%	38.06%	45.48%	2.21
$\boldsymbol{ARX^{0.3}}$	54.57	12.75%	24.39%	37.14%	2.04
RU Generalization Algorithm	5.92	18.34%	13.83%	32.17%	13,425.31

Table 28 indicates that as the maximum average risk of the privacy model in ARX increases, the total FADR increases and the total *Loss* decreases. Since we have defined the maximum information loss, it is possible that the normalized total *Loss* exceeds the 100 percent. That means the total information loss exceeds the maximum value the user defined. Comparing the total FADR and total *Loss* of the generalized datasets from ARX and our algorithm, we can see that our algorithm produces the highest total disclosure risk and the lowest total information loss. Although the total disclosure risk of our algorithm is highest, the maximum of records' FADR values in our obtained generalized dataset is 5.92, which is the lowest maximum record's FADR value compared to the ARX generalized datasets. This indicates the effectiveness of our algorithm in targeting the

records at high risk of disclosure and reducing the disclosure risk at the record level. Besides, we can see that ARX does not prioritize the high-risk records by comparing the two privacy models of 0.01 and 0.1 maximum average risk thresholds. The former outputs higher maximum record's FADR value while incurring lower total FADR.

To evaluate the generalized datasets based on both disclosure risk and information loss at the same time, we considered the objective function defined in Chapter 4, as the summation of normalized total FADR and normalized total *Loss*. Table 28 shows that our algorithm outputs the minimum of the objective function, compared to the ARX outputs.

The execution time of the algorithms are shown in Table 28. ARX anonymization tool is highly efficient in time, compared to our algorithm, because of the pruning strategy that is used in global recoding algorithm. As described before, the privacy models in ARX have monotonicity property, which enables the pruning strategy on the generalization lattice. The privacy model we used in our experiments is average re-identification risk model, which only measures the identity disclosure by considering the inverse frequency of the known tuples. However, our RU Generalization algorithm uses the FADR measure which also considers the attribute disclosure attacks and therefore is not monotone in the generalization lattice and cannot employ the pruning strategy. Our RU Generalization algorithm is slower because of our greedy approach of reducing the disclosure risk of high-risk records by incurring low information loss, at each iteration. The high execution time of our algorithm results in a considerable decrease in information loss and significant reduction in maximum disclosure risk.

Figure 20 compares the distribution of records' FADR values in the original dataset, the generalized dataset obtained from our RU Generalization algorithm, and ARX with 0.3 maximum average re-identification risk model. It shows that most records in the original dataset have high disclosure risk. With ARX, the majority of records converted to low risk records (FADR between 0 and 0.1). However, a few records (76 records) still have high disclosure risk (FADR between 10 and 70). With our generalization algorithm, no records with disclosure risk of more than 10 exist, and the majority of records are converted to have FADR values between 0.1 and 0.2.

Figure 21 illustrates the percentage of original values in each attribute that are remained intact after generalization, with both ARX and our RU Generalization algorithm. The *Gender* attribute has not been generalized with either ARX or our RU Generalization algorithm. The attributes that have been generalized are *Age*, *Work class*, *Education*, *Occupation*, *Race*, and *Country*. These are the attributes that have been classified as quasi-identifiers in ARX, and appeared in the known sets of our RU Generalization algorithm. Figure 21 shows that the generalized dataset obtained from our RU Generalization algorithm better preserves the original values of *Age*, *Education*, *Occupation*, and *Country* attributes, compared to the generalized datasets obtained from ARX trials. *Race* attribute has not been generalized with ARX but our RU Generalization algorithm generalizes 0.54% of the *Race* values. *Work class* attribute has the highest number of non-generalized values in ARX trial of 0.1 maximum average record's risk parameter.

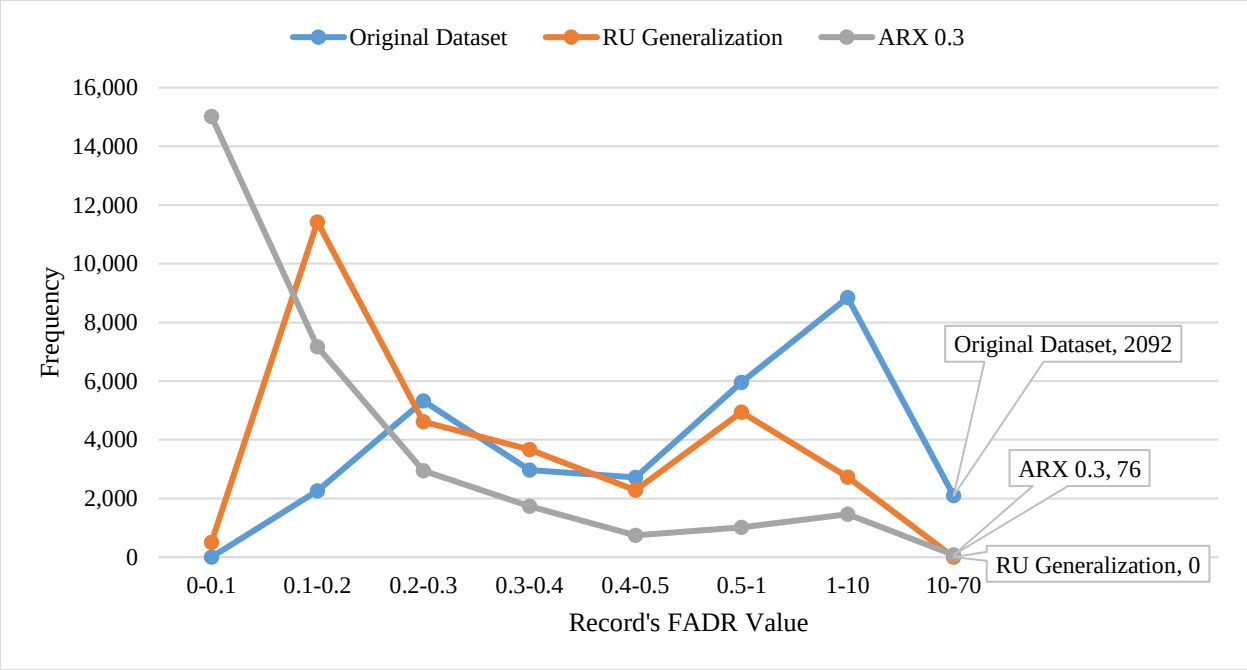


Figure 20. Distribution of records' FADR values on original dataset and generalized datasets obtained from our algorithm and ARX

Figure 21 indicates the percentage of attribute values that are not generalized. To see the details on the generalized values, Table 30-35 show the distribution of attribute values in the generalized datasets from ARX and our RU Generalization algorithm. Since our algorithm and the ARX implementation are both using local recoding generalization, we can see that the attributes in the generalized datasets have values from multiple levels of generalizations.

Table 30 shows how the *Age* attribute values are generalized. ARX follows a generalization hierarchy for *Age* attribute, shown in Table 29. However, our RU Generalization algorithm does not use generalization hierarchy trees for the numeric attributes. Our RU Generalization algorithm generalizes the age value to the range of values of the group that is going to be generalized. Therefore, as Table 30 shows, our algorithm creates more categories of age generalized values compared to ARX that has specific levels of generalizations.

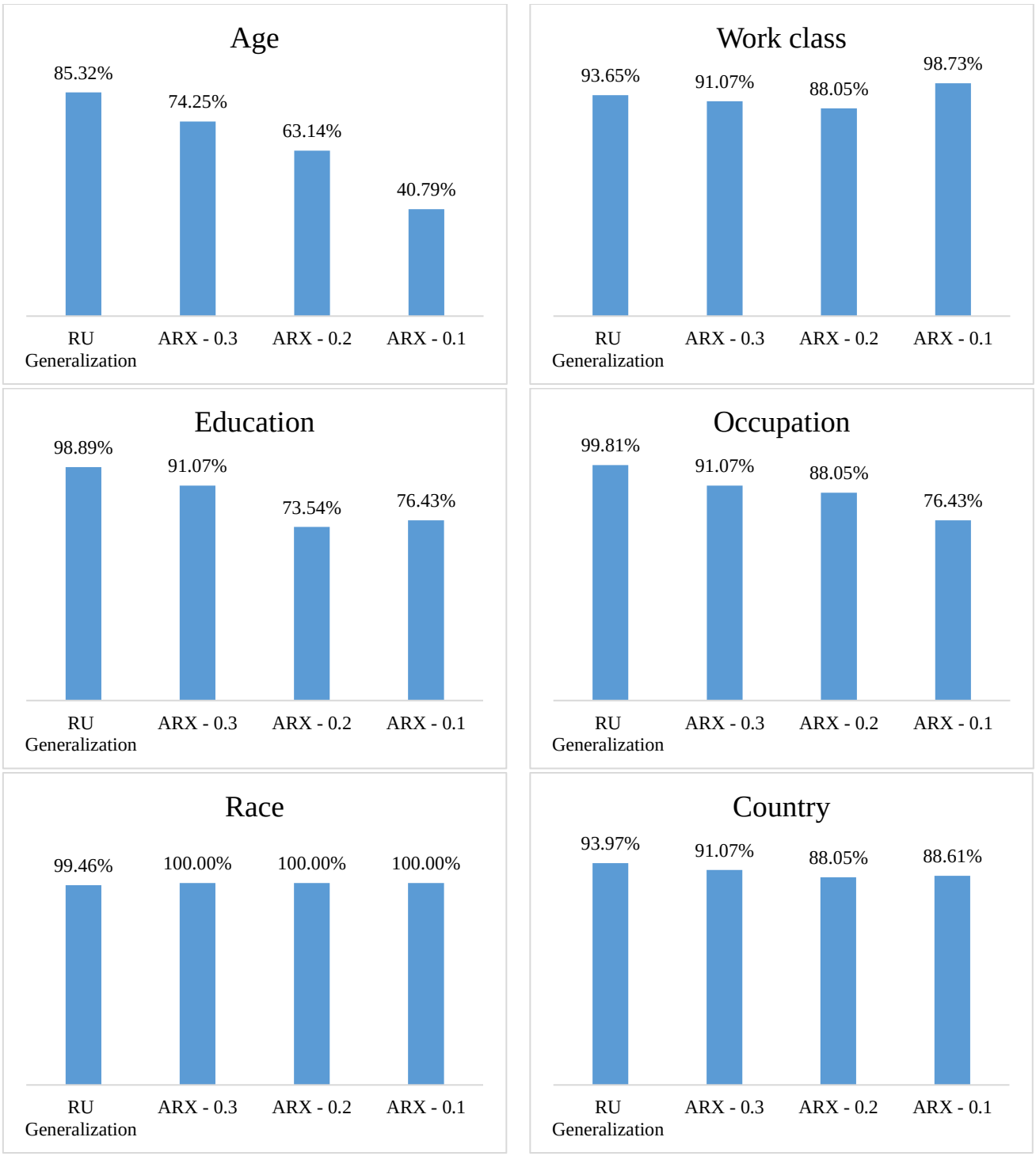


Figure 21. Percentage of original values in each attribute that are preserved in the generalized datasets obtained from our RU Generalization algorithm and from the ARX Anonymization Tool

Table 29. Generalization hierarchy tree for *Age* attribute used in ARX

Level 1	**Level 2**	**Level 3**	**Level 4**
[1, 5]	[1, 10]	[1, 20]	*
[6, 10]	[1, 10]	[1, 20]	*
[11, 15]	[11, 20]	[1, 20]	*
[16, 20]	[11, 20]	[1, 20]	*
[21, 25]	[21, 30]	[21, 40]	*
[26, 30]	[21, 30]	[21, 40]	*
[31, 35]	[31, 40]	[21, 40]	*
[36, 40]	[31, 40]	[21, 40]	*
[41, 45]	[41, 50]	[41, 60]	*
[46, 50]	[41, 50]	[41, 60]	*
[51, 55]	[51, 60]	[41, 60]	*
[56, 60]	[51, 60]	[41, 60]	*
[61, 65]	[61, 70]	[61, 80]	*
[66, 70]	[61, 70]	[61, 80]	*
[71, 75]	[71, 80]	[61, 80]	*
[76, 80]	[71, 80]	[61, 80]	*
[81, 85]	[81, 90]	[81, 100]	*
[86, 90]	[81, 90]	[81, 100]	*
[91, 95]	[91, 100]	[81, 100]	*
[96, 100]	[91, 100]	[81, 100]	*

Table 31 shows that 4.94% of the *Work class* attribute values, in the generalized dataset obtained from our RU Generalization algorithm, are generalized to *Government, Non-Government, Unemployed* categories while 1.38% of the values are suppressed.

Table 32 shows that about 1% of the *Education* attribute values, in the generalized dataset obtained from our RU Generalization algorithm, are generalized to *Primary School, High School, Undergraduate, Graduate, Professional Education* categories, and less than one percent are either generalized to *Primary Education, Secondary Education, Higher Education* categories or have been suppressed.

Table 30. Generalized values for *Age* attribute in the obtained generalized datasets

Generalized Values			
RU Generalization Algorithm	**ARX 0.3**	**ARX 0.2**	**ARX 0.1**
[17, 20], [20, 32], [26, 36], [32, 75], [38, 47], [48, 68]			
[17, 22], [20, 34], [26, 47], [33, 37], [38, 51], [48, 90]			
[17, 23], [20, 43], [27, 42], [33, 40], [38, 52], [49, 66]			
[17, 26], [20, 55], [27, 43], [33, 51], [38, 59], [49, 90]			*
[17, 29], [21, 28], [27, 47], [33, 52], [38, 63], [50, 61]			[1, 20]
[17, 31], [21, 31], [27, 53], [33, 53], [40, 49], [50, 68]		*	[16, 20]
[17, 33], [21, 34], [27, 77], [33, 54], [40, 56], [51, 65]		[1, 20]	[21, 25]
[17, 34], [21, 42], [28, 33], [33, 59], [40, 58], [52, 70]		[11, 20]	[21, 40]
[17, 35], [21, 46], [28, 43], [33, 61], [40, 62], [53, 90]		[21, 30]	[26, 30]
[17, 36], [22, 44], [28, 47], [33, 66], [41, 68], [55, 64]	[1, 20]	[21, 40]	[31, 35]
[17, 47], [22, 49], [28, 49], [33, 90], [43, 60], [55, 66]	[21, 40]	[31, 40]	[36, 40]
[17, 71], [23, 29], [28, 56], [34, 55], [44, 75], [55, 69]	[41, 60]	[41, 50]	[41, 45]
[17, 75], [23, 37], [29, 30], [34, 57], [45, 59], [55, 71]	[61, 80]	[41, 60]	[41, 60]
[17, 77], [23, 40], [29, 31], [35, 71], [45, 69], [56, 66]	[81, 100]	[51, 60]	[46, 50]
[17, 80], [23, 41], [29, 43], [36, 47], [46, 57], [57, 71]		[61, 70]	[51, 55]
[18, 29], [23, 43], [29, 45], [36, 50], [46, 60], [61, 90]		[61, 80]	[56, 60]
[18, 31], [23, 49], [30, 40], [36, 52], [46, 66], [62, 90]		[71, 80]	[61, 65]
[18, 32], [23, 58], [31, 34], [36, 55], [47, 59], [65, 68]		[81,100]	[61, 80]
[18, 35], [24, 27], [31, 77], [36, 57], [47, 61], [65, 90]			[66, 70]
[19, 26], [24, 62], [32, 33], [37, 41], [47, 70], [67, 90]			[81,100]
[19, 29], [25, 40], [32, 43], [38, 44], [48, 62], [70, 90]			
[19, 33], [25, 41], [32, 48], [38, 45], [48, 65], [71, 90]			
[72, 90]			

Table 33 shows that in the generalized dataset obtained from our RU Generalization algorithm, 0.19% of the *Occupation* attribute values are generalized to *Technical*, *Non-Technical*, *Other* categories while no values are suppressed.

Table 31. Relative frequency of *Work class* attribute values in different generalization levels

Generalization Levels of Attribute "Work Class"	Level 0 {original values}	Level 1 {Government, Non-Government, Unemployed}	Level 2 {*}
RU Generalization Algorithm	93.65%	4.94%	1.38%
0.3 max average re-identification risk	91.07%	8.93%	0.00%
0.2 max average re-identification risk	88.05%	11.95%	0.00%
0.1 max average re-identification risk	98.73%	0.00%	1.27%

Table 32. Relative frequency of *Education* attribute values in different generalization levels

Generalization Levels of Attribute "Education"	Level 0 {original values}	Level 1 {Primary School, High School, Undergraduate, Graduate, Professional Education}	Level 2 {Primary Education, Secondary Education, Higher Education}	Level 3 {*}
RU Generalization Algorithm	98.89%	1.01%	0.02%	0.08%
0.3 max average re-identification risk	91.07%	0.00%	8.93%	0.00%
0.2 max average re-identification risk	73.54%	25.11%	0.00%	1.35%
0.1 max average re-identification risk	76.43%	12.17%	10.12%	1.27%

Table 33. Relative frequency of *Occupation* attribute values in different generalization levels

Generalization Levels of Attribute "Occupation"	Level 0 {original values}	Level 1 {Technical, Non-Technical, Other}	Level 2 {*}
RU Generalization Algorithm	99.81%	0.19%	0.00%
0.3 max average re-identification risk	91.07%	8.93%	0.00%
0.2 max average re-identification risk	88.05%	10.60%	1.35%
0.1 max average re-identification risk	76.43%	22.30%	1.27%

Table 34. Relative frequency of *Race* attribute values in different generalization levels

Generalization Levels of Attribute "Race"	Level 0 {original values}	Level 1 {*}
RU Generalization Algorithm	99.46%	0.54%
0.3 max average re-identification risk	100.00%	0.00%
0.2 max average re-identification risk	100.00%	0.00%
0.1 max average re-identification risk	100.00%	0.00%

Table 34 shows that our RU Generalization algorithm suppressed 0.54% of the *Race* attribute values. Table 35 shows that in the generalized dataset obtained from our RU Generalization algorithm, 0.47% of the *Country* attribute values are generalized to *Africa*, *Asia*, *Europe*, *North America*, *South America* categories, and 5.56% of values are suppressed.

Table 35. Relative frequency of *Country* attribute values in different generalization levels

Generalization Levels of Attribute "Country"	Level 0 {original values}	Level 1 {Africa, Asia, Europe, North America, South America}	Level 2 {*}
RU Generalization Algorithm	93.97%	0.47%	5.56%
0.3 max average re-identification risk	91.07%	8.93%	0.00%
0.2 max average re-identification risk	88.05%	11.95%	0.00%
0.1 max average re-identification risk	88.61%	10.12%	1.27%

6 CONCLUSIONS AND FUTURE WORK

In this study, we developed a novel privacy disclosure risk measure, at the records level, named FADR, as a combined measure of identity and attribute disclosure measure. FADR considers all possible external knowledge and disclosure target and provides the flexibility in modeling different adversaries. A pruning algorithm is developed to handle the calculation efficiency of FADR measure. A set of experiments have been conducted to show the effectiveness of the pruning algorithm on the efficiency of FADR calculation and the robustness of FADR measure to the small changes on the input parameters.

In addition, we developed RU Generalization algorithm to obtain an optimized generalized dataset. Unlike the anonymization algorithms in the literature that satisfy a pre-defined privacy level by incurring the minimum information loss, our RU Generalization algorithm aims at minimizing the combination of both disclosure risk and information loss. Our algorithm is a greedy heuristic algorithm that targets the records at high disclosure risk and applies generalization on such records with the lowest information loss. We used our developed FADR measure as the disclosure risk metric in our RU Generalization algorithm since FADR enables our generalization algorithm to consider different adversaries and address both identity and attribute disclosure attacks.

We compared our RU Generalization algorithm with the Flash benchmark generalization algorithm that is implemented in ARX anonymization tool. Through a set of experiments, we have shown that our RU Generalization algorithm outperforms the Flash algorithm with respect to significant reduction in the maximum record's disclosure risk and total information loss. Flash algorithm has shown better efficiency than our algorithm, due to the pruning strategy that is applicable for in ARX privacy models. However, the privacy model of disclosure risk in ARX only considers the identity disclosure, whereas our FADR measure addresses the attribute disclosure attack as well.

Our developed FADR measure can be extended to include different privacy requirements. One future work direction of this study is to address homogeneity and similarity attacks in FADR measure. The consequence term in FADR can add penalties to records being threatened by such attacks. In addition, our RU Generalization algorithm requires improvements in efficiency. We can integrate the generalization lattice and pruning strategy of the Flash algorithm into our RU Generalization algorithm to reduce the computation complexity of our algorithm.

REFERENCES

1. *HIPAA, "Health Insurance Portability and Accountability Act", available at https://www.hhs.gov/hipaa/index.html.*

2. Sweeney, L., *k-anonymity: A model for protecting privacy.* International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002. 10(05): p. 557-570.

3. Manning, A.M., D.J. Haglin, and J.A. Keane, *A recursive search algorithm for statistical disclosure assessment.* Data Mining and Knowledge Discovery, 2008. 16(2): p. 165-196.

4. Abril, D., G. Navarro-Arribas, and V. Torra, *Improving record linkage with supervised learning for disclosure risk assessment.* Information Fusion, 2012. 13(4): p. 274-284.

5. Abril, D., G. Navarro-Arribas, and V. Torra, *Choquet integral for record linkage.* Annals of Operations Research, 2012. 195(1): p. 97-110.

6. Torra, V., G. Navarro-Arribas, and D. Abril, *Supervised learning for record linkage through weighted means and OWA operators.* Control and Cybernetics, 2010. 39: p. 1011-1026.

7. Abril, D., V. Torra, and G. Navarro-Arribas, *Supervised learning using a symmetric bilinear form for record linkage.* Information Fusion, 2015. 26: p. 144-153.

8. Muralidhar, K. and J. Domingo-Ferrer. *Rank-based record linkage for re-identification risk assessment*. in *International Conference on Privacy in Statistical Databases*. 2016. Springer.

9. Domingo-Ferrer, J., S. Ricci, and J. Soria-Comas. *Disclosure risk assessment via record linkage by a maximum-knowledge attacker*. in *Privacy, Security and Trust (PST), 2015 13th Annual Conference on*. 2015. IEEE.

10. Nin, J., J. Herranz, and V. Torra, *Using classification methods to evaluate attribute disclosure risk.* Modeling Decisions for Artificial Intelligence, 2010: p. 277-286.

11. Herranz, J., et al., *Classifying data from protected statistical datasets.* computers & security, 2010. 29(8): p. 875-890.

12. Torra, V., *Privacy Models and Disclosure Risk Measures*, in *Data Privacy: Foundations, New Developments and the Big Data Challenge*. 2017, Springer International Publishing: Cham. p. 111-189.

13. Machanavajjhala, A., et al., *l-diversity: Privacy beyond k-anonymity.* ACM Transactions on Knowledge Discovery from Data (TKDD), 2007. 1(1): p. 3.

14. Soria-Comas, J., et al., *t-closeness through microaggregation: Strict privacy with enhanced utility preservation.* IEEE Transactions on Knowledge and Data Engineering, 2015. 27(11): p. 3098-3110.

15. Motwani, R. and Y. Xu. *Efficient algorithms for masking and finding quasi-identifiers*. in *Proceedings of the Conference on Very Large Data Bases (VLDB)*. 2007.

16. Wang, L. and Q. Zhu, *Utility-based anonymisation for dataset with multiple sensitive attributes.* International Journal of High Performance Computing and Networking, 2016. 9(5-6): p. 401-408.

17. Zhang, Q., et al. *Aggregate query answering on anonymized tables*. in *2007 IEEE 23rd International Conference on Data Engineering*. 2007. IEEE.

18. Li, Y., et al. *The Hardness of (ε, m)-anonymity*. in *International Conference on Web-Age Information Management*. 2013. Springer.

19. Li, N., T. Li, and S. Venkatasubramanian. *t-closeness: Privacy beyond k-anonymity and l-diversity*. in *2007 IEEE 23rd International Conference on Data Engineering*. 2007. IEEE.

20. Liu, Q., H. Shen, and Y. Sang. *A Privacy-Preserving Data Publishing Method for Multiple Numerical Sensitive Attributes via Clustering and Multi-sensitive Bucketization*. in *Parallel Architectures, Algorithms and Programming (PAAP), 2014 Sixth International Symposium on*. 2014. IEEE.

21. Han, J., et al., *SLOMS: a privacy preserving data publishing method for multiple sensitive attributes microdata.* Journal of Software, 2013. 8(12): p. 3096-3104.

22. Nergiz, M.E., C. Clifton, and A.E. Nergiz, *Multirelational k-anonymity.* IEEE Transactions on Knowledge and Data Engineering, 2009. 21(8): p. 1104-1117.

23. Terrovitis, M., N. Mamoulis, and P. Kalnis, *Privacy-preserving anonymization of set-valued data.* Proceedings of the VLDB Endowment, 2008. 1(1): p. 115-125.

24. He, Y. and J.F. Naughton, *Anonymization of set-valued data via top-down, local generalization.* Proceedings of the VLDB Endowment, 2009. 2(1): p. 934-945.

25. Truta, T.M. and B. Vinay. *Privacy Protection: p-Sensitive k-Anonymity Property*. in *ICDE workshops*. 2006. Citeseer.

26. Wong, R.C.-W., et al. *(α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing*. in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006. ACM.

27. Rubner, Y., C. Tomasi, and L.J. Guibas, *The earth mover's distance as a metric for image retrieval.* International journal of computer vision, 2000. 40(2): p. 99-121.

28. Li, N., T. Li, and S. Venkatasubramanian, *Closeness: A new privacy measure for data publishing*. IEEE Transactions on Knowledge and Data Engineering, 2010. 22(7): p. 943-956.

29. Li, J., Y. Tao, and X. Xiao. *Preservation of proximity privacy in publishing numerical sensitive data*. in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 2008. ACM.

30. Loukides, G. and J. Shao, *Preventing range disclosure in k-anonymised data*. Expert Systems with Applications, 2011. 38(4): p. 4559-4574.

31. Loukides, G., A. Tziatzios, and J. Shao. *Towards preference-constrained k-anonymisation*. in *International Conference on Database Systems for Advanced Applications*. 2009. Springer.

32. Loukides, G. and J. Shao. *Capturing data usefulness and privacy protection in k-anonymisation*. in *Proceedings of the 2007 ACM symposium on Applied computing*. 2007. ACM.

33. Liu, Q., H. Shen, and Y. Sang. *A Privacy-Preserving Data Publishing Method for Multiple Numerical Sensitive Attributes via Clustering and Multi-sensitive Bucketization*. in *2014 Sixth International Symposium on Parallel Architectures, Algorithms and Programming*. 2014. IEEE.

34. Wang, H., et al. *(k, ε)-Anonymity: An anonymity model for thwarting similarity attack*. in *Granular Computing (GrC), 2013 IEEE International Conference on*. 2013. IEEE.

35. Nergiz, M.E., M. Atzori, and C. Clifton. *Hiding the presence of individuals from shared databases*. in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. 2007. ACM.

36. Nergiz, M.E. and C. Clifton, *δ-presence without complete world knowledge*. IEEE Transactions on Knowledge and Data Engineering, 2010. 22(6): p. 868-883.

37. Fung, B., et al., *Privacy-preserving data publishing: A survey of recent developments*. ACM Computing Surveys (CSUR), 2010. 42(4): p. 14.

38. Loukides, G., A. Gkoulalas-Divanis, and B. Malin, *Anonymization of electronic medical records for validating genome-wide association studies*. Proceedings of the National Academy of Sciences, 2010. 107(17): p. 7898-7903.

39. Gkoulalas-Divanis, A., G. Loukides, and J. Sun, *Publishing data from electronic health records while preserving privacy: a survey of algorithms*. Journal of biomedical informatics, 2014. 50: p. 4-19.

40. Li, T., et al., *Slicing: A new approach for privacy preserving data publishing.* IEEE transactions on knowledge and data engineering, 2012. 24(3): p. 561-574.

41. LeFevre, K., D.J. DeWitt, and R. Ramakrishnan. *Incognito: Efficient full-domain k-anonymity*. in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. 2005. ACM.

42. LeFevre, K., D.J. DeWitt, and R. Ramakrishnan. *Mondrian multidimensional k-anonymity*. in *22nd International Conference on Data Engineering (ICDE'06)*. 2006. IEEE.

43. LeFevre, K., D.J. DeWitt, and R. Ramakrishnan. *Workload-aware anonymization*. in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006. ACM.

44. Li, J., et al. *Achieving k-anonymity by clustering in attribute hierarchical structures*. in *International Conference on Data Warehousing and Knowledge Discovery*. 2006. Springer.

45. Ghinita, G., et al. *Fast data anonymization with low information loss*. in *Proceedings of the 33rd international conference on Very large data bases*. 2007. VLDB Endowment.

46. Yang, J., et al. *A data anonymous method based on overlapping slicing*. in *Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on*. 2014. IEEE.

47. El Emam, K. and F. Dankar, *Re-identification risk in de-identified databases containing personal information*. 2012, Google Patents.

48. El Emam, K., *Risk-based de-identification of health data.* IEEE Security & Privacy, 2010(3): p. 64-67.

49. Iyengar, V.S. *Transforming data to satisfy privacy constraints*. in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002. ACM.

50. Xu, J., et al. *Utility-based anonymization using local recoding*. in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006. ACM.

51. *ARX – Data Anonymization Tool, available at http://arx.deidentifier.org/.*

52. Prasser, F. and F. Kohlmayer, *Putting statistical disclosure control into practice: The ARX data anonymization tool*, in *Medical Data Privacy Handbook*. 2015, Springer. p. 111-148.

53. Kohlmayer, F., et al. *Flash: efficient, stable and optimal k-anonymity*. in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*. 2012. IEEE.

VITA

Marmar Orooji received her Bachelor of Science in Electrical Engineering from Shahed University, Tehran, Iran, in 2013. Thereafter, she started her doctoral program in the college of engineering at Louisiana State University (LSU). During her Ph.D. program, she earned two Master of Science degrees from LSU; one in Engineering Science with concentration in Information Technology & Engineering in 2017, and the other in Computer Science in 2019. During her graduate studies, she served as a teaching assistant and primary instructor for four semesters. In addition, she worked as a graduate assistant at LSU Social Research & Evaluation Center (SREC) as a database administrator and data scientist for four years. She is a candidate for the Doctor of Philosophy degree in Engineering Science with concentration in Information Technology & Engineering (ITE) under supervision of Dr. Gerald Knapp. The degree will be conferred at the Summer commencement 2019. Upon her graduation, she will start working at Rice University as a teaching-track faculty in Computer Science department.