Assalamu alaikum I am Rashik Rahman. Up until now we have discussed about the problem in hand. So now lets discuss about the solve. This paper is about a new anonymizing technique using cell generalization and merging algorithm. We want to increase the possibility of false matches by grouping the attributes in equivalence class and use permutation to break the correlation between attributes. Specifically, the process consists of fake tuples, attribute separation, tuple separation and cell generalization.

#####Fake Tuple#####

A published table with generalization has less data utility than the published table with faketuples. The presence of fake tuples does not have any effect on the utility of the published dataset,

but will increase the probability of false matches during a composition attack on the published table [25].

In our anonymization method, we create n (n=1,2,3,...) fake tuples with the same QI values as in the

original table and assigned sensitive values to them based on the sensitive value distribution in the

initial dataset.

#####Attribute Separation#####

For attribute separation, related attributes are arranged in a subset, such that each attribute

belongs to one subset. Each grouped subset is called a column.Related attributes are grouped by measuring the association

between attributes. The widely usedmethod to measure the association between categorical attributes is the mean square

contingency coefficient [25,42].We arrange attributes to bring the highly correlated attributes in the same column.

For the data utility and privacy, this approach performs well. In terms of data utility, grouping highly correlated

attributes protect the relationships among those attributes.With regard to privacy, the association

between correlated attributes reduce the identification risk. Since the association between uncorrelated

attribute values are not common among the dataset and hence more distinctive, it is suitable to break

the relation between uncorrelated attributes to preserve the user privacy. That way, when an adversary examines the

intersection of two or more published datasets, the resulting intersection data will lead to more false matches,

reducing the adversary’s confidence in breaching data privacy

#####Tuple Separation#####

This entails generating different subsets of T, such that each tuple is assigned to one subset. Every

subset of tuples is called an equivalence classThe tuple separation operation divides the records horizontally into a number

of partitions, called buckets or equivalence classes. The Mondrian [44] algorithm is applied, and it follows the

top-down approach without generalization feature to separate tuples in the equivalence classes. Within

each equivalence class, the values in each column are randomly permuted to break the cross-column

associations. Therefore, the tuple separation will minimize the linkage between the sensitive values

with the QI values which will reduce the adversary’s confidence in linking with the sensitive values.

#####Cell Generalization#####

A cell is the cross-section of a column and a row. In our problem definition, a cell consists of

a column and an equivalence class. Given a microdata table T, a column Cc and an equivalence

class Ee make a cell CE(i,j). A cell generalization for CE(i,j) generalizes each attribute value of CE(i,j)

to satisfy privacy requirements.

In the anonymization, the attribute values in the equivalence class will be shuffled randomly

to break the association of each tuple, in order to increase personal privacy. Random shuffling will

break the association of the tuple but it will create some invalid records and, in some cases, it may

increase the likelihood of privacy breach for a particular set of sensitive values [22]. For these special

circumstances, we have introduced cell generalization to enhance the privacy of the equivalence class.

Because cell generalization does not generalize the whole equivalence class, it allows better data utility

than column generalization or full generalization of the table.

Cell generalization increases the probability of false matches of the attributes

and it certainly increases the data utility of the published dataset.

#####Anonymization Algorithm#####

Our anonymization algorithm performs the anonymization process as follows:

It creates n fake tuples at line 2 and adds them to the original microdata table T. It maintains two

data structures: a queue of equivalence classes Q and a set of anonymized equivalence classes ET.

Initially, Q contains only one equivalence class, and ET is empty. In each iteration (lines 4 to 10),

the anonymization algorithm removes an equivalence class from Q and breaks the equivalence class

into two equivalence classes according to the Mondrian [44] criterion. In line 7, privacy is checked by

the Privacy-Check algorithm and the two equivalence classes are appended at the end of the queue

Q (for more breaks of the equivalence class this is in line 8). If the equivalence class can no longer be

broken, then the anonymization algorithm puts the equivalence class into ET in line 9. Finally, in line

12, the anonymized table T is published.

#####Algorithm for Privacy Checking#####

Our privacy checking algorithm assures the privacy requirement R in each equivalence class. In the

anonymization, column values are permuted randomly to break cross-column associations. There is a

possibility of creating some invalid records [22] or incompatible tuples in the process. In line 2, tuple

incompatibility is checked as in [22]. If there are incompatible tuples, we generalize the particular cell

values to satisfy k-anonymity. In line 5 we check the l-diversity privacy requirement as in slicing [25].

#####Discussion on the Merging Method#####

Now I illustrate how the Merging method is able to protect the anonymization

table from composition attack while increasing the data utility and maintaining l-diversity in the

intersection of two published tables. Tables 6 and 7 are the published tables from Hospitals A and B.

Consider a tuple t with QI values (22, M, 47905), where tuple t visits both hospitals for medication.

We can create a search for the QI values of tuple t in the intersection shown in Table 8. To determine

the QI and sensitive values of t, its matching equivalence class is examined. In the first column,

the attributes (Age, Sex) have the values (22,\*), and in the second column, Zipcode has the value 47905

in the first equivalence class. Therefore, we can say the person may exist in the first equivalence class.

Because the Sex attribute value is generalized, there is a possibility that the individual could be female

even though the attribute Sex is considered to be male; the QI values (22, M, 47905) are linked to three

sensitive values. Based on negative association rules [22], a male cannot suffer from breast cancer,

so we deduct one more value and there are two values, satisfying l-diversity [25].

#####Data Set#####

Data sets are described in Table 9.

In our experiments, we extracted two independent datasets from the Adult dataset (i) Occupation

and (ii) Education. The Adult dataset has 48842 tuples with six QI attribute values: Age, Sex, Marital

status, Work class, Relationship, and Salary. Occupation is used as the sensitive attribute value for the

Occupation dataset, and Education for the Education dataset.

#####Experimental Analysis#####

In the analysis part we present experiments on real-world datasets. The experiments are divided

into two parts: the first part was designed to test the effectiveness of the proposed anonymization

algorithm against composition attack, in comparison with the e e DP [41], Hybrid [2], Probabilistic [23],

Composition [24] and Mondrian [44] methods. Our experimental results show that the Merging method

also provides smaller privacy risks for the composition attacks. The results of this experiment are

presented in the figs below.Composition attacks were measured by calculating privacy risk for all pairs of the extracted

dataset with identical overlapping records. In the Merging method, the false matches will be increased

because of l distinct sensitive values linked with the QI values, and it will decrease the privacy risk.

Figures 1–4 present the experimental results on the Occupation and Education datasets,

respectively. They illustrate the privacy risk resulting from different anonymization techniques.

Privacy risk indicates how confidently an adversary can learn sensitive values of a user from the

multiple independent datasets. e e DP [41] provides the lowest privacy risk for composition attacks

among all the compared methods. Privacy risk gets smaller by increasing the false matches in the

published datasets. The breaking of cross-column relation increases the probability of false matches

in the anonymized datasets by the Merging method. As reported by the privacy risk shown in

the result, Merging yields a lower probability of inferring the user’s private information than the

Probabilistic [23], Composition [24] and Mondrian [44] methods. It has almost identical privacy risk

to that of the Hybrid [2] method. Therefore, we can say Merging also reduces the probability of

composition attack on published datasets.

In the second part, we evaluated the effectiveness of our Merging method in preserving data

utility, as compared to the same set of competing methods. The experiment demonstrates that the

Merging method preserves more data utility than the other methods. In addition, it has smaller

relative query error and better classification accuracy than the competing methods. we measured the data quality

by the distortion ratio and the aggregate query answering error. In addition, we conducted an experiment on

classifier learning of the published datasets. We observed that the Merging method has lower data loss,

less relative error and better classification accuracy than the all other methods. The results of this

experiment are presented in the figs below.

#####Execution Time#####

We measure the scalability of the Merging method by evaluating the computation time to run the

anonymization algorithm. To measure the computation time, we fix l = 6 and increase the dataset sizes

for the execution time. Figure 9 presents the computation time as a function of the number of records.

The results show that the Merging algorithm scales well with the data sizes.