Privacy Preserving in Big Data Sets through Multiple Shuffle

Youyang Qu
School of Information Technology
Deakin University
Victoria, Australia
quyo@deakin.edu.au

Jiyang Xu
School of Information Technology
Deakin University
Victoria, Australia
asjjnwq@deakin.edu.au

Shui Yu
School of Information Technology
Deakin University
Victoria, Australia
syu@deakin.edu.au

ABSTRACT

Big data privacy-preserving has attracted increasing attention of researchers in recent years. But existing models are so complicated and time-consuming that they are not easy to implement. In this paper, we propose a more feasible and efficient model for big data sets privacy-preserving using shuffling multiple attributes(M-Shuffle) to achieve a tradeoff between data utility and privacy. Our strategy is firstly categorize all the records into some groups using K-means algorithm according to the sensitive attributes. Then we choose the columns to be shuffled using entropy. At last we introduce the random shuffle algorithm to our model to break the correlation among the columns of big data sets. Experiments on real-world datasets show that our framework achieves excellent data utility and efficiency while satisfying privacy-preserving.

Keywords

privacy-preserving; K-means; M-Shuffle machenism

1. INTRODUCTION

In the information age, big data is a milestone and leads to sharp changes to modern society. Government agencies, big IT companies and other organizations always publish big data sets for research purpose, for example, the census or medical datasets. But releasing the datasets to public may cause privacy leakage because every record stored in such kind of data sets corresponds to one specific individual. Privacy preserving in big data sets is hence to become a big challenge worldwide.

There are two main branches in privacy research: privacy-preserving data publishing(PPDP) and privacy-preserving data mining(PPDM) [1]. In PPDP, there are several milestones like k-anonymity [2], l-diversity [3], t-closeness [4], and differential privacy [5][6]. The extensions of these models are also widely studied in recent years. The k-anonymity and its extensions are likely to suffer homogeneity attack and background knowledge attack, which makes it very vulnerable. The l-diversity and its extensions sometimes lose more information and lead to a larger utility loss. Although t-closeness offers better privacy protection, it suffers the same challenge of l-diversity in some occasions. Differential privacy offers a theoretical foundation but sometimes it is too strict to implement on real-world systems and the efficiency is another big challenge [7].

However, there are three urgent challenges needs to be solved.

The first challenge is that privacy and data utility seems to be natural antithesis. A tradeoff must be found to preserve privacy of big data sets while the utility of the big data sets should be maintained in a proper level [8]. It's hard to satisfy all the requirements.

The second challenge is that the correlation between the values of the same record, anonymization and generalization could partly decouple the correlation between the values, but the utility will suffer a great loss in this [5]. The time complexities of this kind of methods are very large, which is a big problem for practical usage.

The third challenge occurs when adversaries have too much background knowledge. Background knowledge is a great threat to all kinds of privacy models because an adversary can re-identify a specific person if he gains some information from the released datasets and combines it with the background knowledge of himself [3].

In order to address these challenges, we propose the multiple shuffle model. In the M-Shuffle, we first use K-means to group all the records to K clusters. Then, we group all the attributes and choose some of the attributes to be shuffled using entropy. At last, we shuffle the chosen columns using random shuffle algorithm. By applying the mechanism, we break the correlation among the values of one single record. In this way, an adversary's background knowledge become less useful and the statistics of a single column will not change at all. Using K-means will improve the utility without sacrificing the privacy level. Our model will maintain the statistics of the shuffled columns and provide a desired privacy level with high efficiency.

The contributions of this paper include:

- Firstly, we propose a new privacy model based on shuffling the chosen columns. All records are categorized into proper clusters with k-means, this could preserve the utility without sacrificing the privacy level.
- Secondly, we theoretically analysis the proposed model and build a mathematical model. The correlations be-

ACSW '17, January 31-February 03, 2017, Geelong, Australia

DOI: http://dx.doi.org/10.1145/3014812.3014886

tween values of a record is decoupled properly so that we find a trade-off between privacy and data utility. Experiments on real-world datasets proved our model efficient and practical.

The rest of this paper is organized as follows. Section 2 will illustrate the related work. In section 3, we present the preliminaries and the problem statement. The algorithm of the proposed model and the theoretical privacy and utility analysis are discussed in section 4. Section 5 presents the experiments and the performances of our model, which is followed by conclusions in Section 6.

2. RELATED WORK

With the development of information technology, the big data age has arrived. Its impacts are so pervasive that we can see its implementations on every aspects of daily life, research, or even government functioning. For example, existing big data sets benefit us a lot in biology [9], social science [10], e-commerce, disease-control, and so on. We can easily predict outburst of a infectious disease thorough social networks which are big data sets in essence. In 2011, Doug Laney [11] proposed an early concept of big data in the Gartner report, where big data was defined as large and complex data sets that current computing facilities were not able to handle. This started the research enthusiasm of big data. Big data never asks why while gives the predictions simply, which makes research on it more valuable.

But releasing the big data sets to the public may lead to privacy disclosure, even for research purpose.There are two main kinds of information disclosure: identity disclosure and attributes disclosure [7]. No matter which of these two situations occurs, it will be harmful to individual's privacy and may even cause financial loss. Privacy study has spring up since two decades ago. First, data clustering methods were carried out to privacy-preserving. The first milestone is the k-anonymity model [2]. It was the first model to introduce the data clustering method to privacy protection, which was proposed in 1998. The following one is L-diversity [3], which showed up in 2007. It is an extension of k-anonymity and introduced diversity into data clustering. Then t-closeness was employed in 2010 [4] which also took distribution into consideration. Models based on data clustering advanced the privacy-level of big datasets. Despite of their feasibleness, lack of firm theoretical analysis is always a flaw.

In 2006, another milestone with firm foundation was developed. Dwork [5][6] proposed the differential privacy. Differential privacy is a framework for formalizing privacy in statistical databases introduced in order to protect against these kinds of de-anonymization techniques. After this, extensions like personalized differential privacy frameworks appeared, for example, the work of Li et al. [4][12]. As to privacy measurement, we can see game theory as a very useful tool. Mechanism design [13] has been employed in this field already since 2013.

3. PRELIMINARIES AND PROBLEM STATEMENT

3.1 Basic Notation

In terms of attributes, a given record will be divided into four categories in this model[8]. An example is shown in Table 1.

Table 1: Components of a Big Data Set

EI	Explicit identifier
QI	Quasi-identifier
SA	Sensitive attribute
OA	Other attribute

1)Explicit identifier(EI). The exclusive attribute which precisely identifies an individual or an organization, such as name, TFN, drive license numbers, etc.

2)Quasi-identifier(QI). Quasi-identifier is the attribute with which an observer may re-identify individuals with a high possibility. The attribute will point to unique individuals after gathering with other information, such as gender, postcode, age, and so on.

3)Sensitive attribute(SA). It is the attribute that adversaries are expecting for, such as disease and salary.

4)Other attribute(OA). OA belongs to the meaningless information that will not impact on the privacy-preserving process.

An example is shown in Table 2. In this table, Name donates the EI, while Job, Age, Sex, Zipcode are the QIs, and salary belongs to the sensitive attribute.

Table 2: Raw Data

No	Name	Job	Age	Sex	Zipcode	salary
1	Alice	clerk	23	F	13010	5500
2	Betty	accountant	25	F	13050	5400
3	Bob	barker	30	M	13020	2300
4	Sophia	editor	31	F	13040	2200
5	Lucy	housewife	36	F	13220	1000
6	Steven	manager	39	M	13221	2000
7	Bill	dustman	42	M	13226	3000
8	Paul	wleder	45	M	13228	4000
9	Ellen	prof.	52	F	14850	5000
10	Jack	mechanic	53	M	14862	2000
11	David	manager	61	M	14802	3000
12	Martin	prof.	64	M	14800	1500

3.2 Problem Statement

In this paper, we discuss the popular scenario that all the statistics are published once and for all. Table 2 reveals the structure of the statistics releasing with privacy guarantee. The aim is that we can publish the statistics without data obtainment of the adversaries. Therefore, instead of releasing the original statistics, we implement a proper privacy protection mechanism. Then, a sanitized statistics will be released.

Problem: Given an raw dataset $D = \{D_1, D_2, ..., D_n\}$, we introduce K-means and M-shuffle to release a sanitized version $R = \{R_1, R_2, ..., R_n\}$. This R should satisfy certain privacy level.

4. SYSTEM MODELING AND ANALYSIS

Intuitively, privacy could be measured by how much information an observer could obtain. The observer has two beliefs and the difference between them shows the information obtainment. One belief is prior belief that the observer has some background knowledge of the sensitive attributes. While the other is posterior belief after gaining the data.

The novelty of our method is that we decouple the correlation between the Quasi-identifiers and the sensitive attributes.

In order to balance the privacy preserving and statistical accuracy, there is a trade-off. In our model, the M-shuffle, M denotes the number of shuffled columns to control the privacy level. Our strategy is that we first categorize all the records with K-means algorithm according to the sensitive attributes, which will keep certain privacy level with better utility. Then we introduce the random shuffle algorithm to our model to break the correlation among the columns of the big data set. We first present an overview of the algorithm and then provide details of its operations.

4.1 M-Shuffle Mechanism

Basic notations. Let $T = \{T_1, T_2, ..., T_n\}$ be a table with attributes $A_1, A_2, ..., A_n$. |T| denotes the number of records in table T. Assume that a larger table T_Ω is the superset of table T. Each record of T_Ω represents a specific individual. For instance, if T is a dataset with basic information of citizens, then T_Ω might be from a census bureau of a certain country. Let A denotes the dataset of all the attributes $\{A_1, A_2, ...A_m\}$. Then $t|A_i|$ is regarded as the value of attribute A_i in table T. We have defined Explicit-identifier(EI), Quasi-identifier(QI), Sensitive attribute(SA) and other attribute(OA) above. But in a table, OA is meaningless and we will remove it first. Therefore, A could be illustrated as

$$A = \{EI_1, .., EI_p, QI_1, ...QI_p, SA_1, ..., SA_p\}$$

As the EIs are removed directly from all record, those are not what we will discuss. But as to the QIs, combination of them have a great potential threat to privacy-preserving. In this paper, we split the QIs into three categories. The first category contains all QIs which have strong correlation with the SAs, for example, job and salary. We name it QIA in this paper. The second category has all QIs which have no more than two values of all records, for example, sex. We name it QIB in this paper.

The third category includes all QIs which have weak correlation with or even independent of the SAs, for example, postcode and salary. We name it QIC in this paper. After categorizing the QIs, A could be represented as

$$A = \{EI_1, .., EI_p, QIA_1, ...QIA_p, QIB_1, ...QIB_p, QIC_1, ...QIC_p, SA_1, ..., SA_p\}$$

Let |A| be the number of all the attributes. For most of the datasets, |A| is less than 20.

Assume all the records has SAs, for example, "salary" in Table 2, which we regard it as the sole SA. Apart from "name", the other columns are regarded as the QIs. However, in these scenarios, a proper method is used to find the trade off between privacy and utility. The method can preserve more utility without sacrificing privacy. Our goal is to release a sanitized dataset while we protect it from any adversaries whose purposes are to breach privacy of the sanitized dataset.

We will group QIs into three categories: QIAs, QIBs, ans QICs. As is shown in Table 3. The QIAs have strong correlation to the SAs. Therefore, adversaries will find it hard to identify a specific person with combinations of QIAs. For example, we know all lawyers have good salaries, so we can not identify a person with his or her job and salary. The QIBs have no more than two values, so it may even harder for adversaries to identify a person with combinations of QIBs. Therefore, we believe it is nearly impossible for adversaries to identify a person with QIAs and QIBs if the record number |T| is larger than |QIAs|+|QIBs|. In real-world dataset, it is common that |EIs|+|QIs|+|SAs|<20, but usually there are hundreds of or thousands of records. Therefore, it is natural that |T|>|QIAs|+|QIBs|. For this reason, we will not operate the QIAs and QIBs while QICs will be processed. We use entropy in information theory [14] $H(x)$ to distinguish QIA and QIB from QIC. We have

$$H(x) = -\sum_{i=1}^{n} P(x_i) log_e P(x_i), \quad (1)$$

where x_i denotes the element of the column and $P(x_i)$ represents the possibility of x_i. Larger H(x) means less correlation between the QI and the SA. After the $H(x) > H_0(x)$, where H_0 denotes the threshold of entropy, this column will be grouped to QIC.

Table 3: Groups of QIs

QIA	The QIs has a strong correlation with SA
QIB	The QIs has no more than two values
QIC	The QIs has weak or even on correlation with SA

First, we will categorize all the records. To protect more utility of a given dataset, we use K-means to sort all the records of a dataset according to the sensitivity of the SAs. The method of K-means will be discussed below. We will sort all the records according to the SAs with respect to a parameter K to control the number of subsets. The K should be changed according to different types of SAs. All the records of a given sensitive attribute may not have the same sensitivity. That is how we divide the dataset to several subsets. For example, In Table1, we divide the records into four groups according to salary: {[0, 2000], [2000, 3500], [3500, 5000], [5000, ∞]}. That is because people with high salaries are naturally to have a different sensitivity comparing to those with low salaries. Intuitively, we will operate the records within a subset in order to preserve privacy as much as possible.

The main goal is that find a proper tradeoff between privacy and utility. Therefore, some operations will be performed on QICs. Combinations of QICs will lead to privacy leakage with high probability. The relations among all the values of records account for the leakage. Our method is partly decouple the relations among them. The specific method is as follows.

As known, all the EIs have been removed from T, which makes

$$A^* = \{QIA_1, ...QIA_p, QIB_1, ...QIB_p, QIC_1, ...QIC_p, SA_1, ..., SA_p\}$$

As discussed above, QIA and QIB only cause slightly privacy leakage while QIC will lead to a lot. We will use $shuffle()$ to denote the shuffle function. In our mechanism, we will keep QIAs, QIBs and SA still and then shuffle the data randomly within each column belonging to QIC. The new A^* will be

$$A^* = \{QIA_1, ...QIA_p, QIB_1, ...QIB_p, shuffle(QIC_1), ... shuffle(QIC_p), SA_1, ..., SA_p\}$$

The shuffle algorithm we choose is a modified Fisher-Yates shuffle called random shuffle algorithm. The algorithm will be discussed below. The example of shuffled table corresponding Table 2 to is as Table 4.

Table 4: Shuffled data with proposed privacy mechanism

No	Job	Age	Sex	Zipcode	salary
1	clerk	23	F	13010	5500
9	prof.	52	F	14850	5000
2	accountant	25	F	13050	5400
8	wleder	45	M	13228	4000
7	dustman	42	M	13226	3000
11	manager	61	M	14802	3200
3	barker	30	M	13020	2300
4	editor	31	F	13040	2200
10	mechanic	53	M	14862	2000
6	manager	39	M	13221	2000
12	prof.	64	M	14800	1500
5	housewife	36	F	13220	1000

4.2 System Analysis

4.2.1 Data Clustering Algorithm

K-means is a very useful tool of data clustering, it is first used in signal processing[15]. K-means aims to partition n records into k clusters in which each record belongs to the cluster with the nearest mean. Iterative update is used in the most common algorithm. The algorithm, k-means, is named because of its extensive existence. It is also referred to as Lloyd's algorithm, particularly in computer science community.

There are two steps to implement this algorithm. Let Mean be the initial set of means and $Mean = \{m_1, m_2, ..., m_k\}$. First, it is the assignment step. In this procedure, assign each record to the cluster whose mean yields the least within-cluster sum of squares. Since the sum of squares is the squared Euclidean distance, this is intuitively the "nearest" mean. Mathematically, this means partitioning the observations according to the Voronoi diagram generated by the means.

$$S_i^{(t)} = x_p : \parallel x_p - m_i^{(t)} \parallel^2 \leqslant \parallel x_p - m_j^{(t)} \parallel^2 \forall j, 1 \leqslant j \leqslant k \tag{2}$$

Then, it is the update step. We calculate the new means to be the barycenter of the records in the new groups.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \subset S_i^{(t)}} x_j \tag{3}$$

Since the arithmetic mean is a least-squares estimator, this also minimizes the within-cluster sum of squares (WCSS) objective. The algorithm has converged when the assignments no longer change. Since both steps optimize the WCSS objective, and there only exists a finite number of such partitionings, the algorithm must converge to a (local) optimum.

The k-means data clustering algorithm is introduced to our mechanism to assign records to the nearest group by range. It has the goal to minimize the WCSS objective, and it is assigned by least sum of squares. The algorithm is often presented as assigning objects to the nearest cluster by distance. The standard algorithm aims at minimizing the WCSS objective, and thus assigns by "least sum of square", which is exactly equivalent to assigning by the smallest Euclidean distance.

Algorithm 1 K-means clustering algorithm

```
Initialization:
Let n be the number of clusters
Let S be the set of feature vectors (|S| is the size of the set)
Let A be the set of associated clusters for each feature vector
Let sim(x,y) be the similarity function
Let c[n] be the vectors for our clusters
Let S' = S
for i = 1; i ⩽ n; i++ do
  j = rand(|S'|)
  c[n] = S'[j]
  S' = S' − {c[n]}
end for
for i = 1; i ⩽ |S|; i++ do
  A[i] = argmax(j = 1ton){sim(S[i], c[j]})
end for
Let change = true
while change do
  change = false
  for i = 1; i ⩽ |S|; i++ do
    a = argmax(j = 1ton{sim(S[i], c[j])}
    if a! = A[i] then
      A[i] = a
      change = true
    end if
  end for
  if change then
    for i = 1; i ⩽ n; i++ do
      mean, count = 0
      for j = 1; j ⩽ |S|; j++ do
        if A[j] == i then
          mean = mean + S[j]
          count = count + 1
        end if
      end for
      c[i] = mean/count
    end for
  end if
end while
```

The time complexity is satisfactory. If k and d are constant values, then the time complexity can be presented as

$$O(n^{dk+1} \log n), \tag{4}$$

where n=|T| denotes the number of records while d denotes the dimension.

Because of the upstate process, iteration times will increase faster and faster which will lead larger time-consuming. Therefore, we find a pratical experience that the target dataset after 1,000-times iteration and the one after 10,000-times iteration are 97% the same. In our experiments, we set 1,000 as the maximum iteration times to improve the efficiency.

4.2.2 Shuffle Algorithm

The Fisher-Yates shuffle is a very popular algorithm which is an in-place shuffle[16]. That means instead of creating a

new shuffled copy of the records, it shuffles the records of a table in place. If the table to be shuffled is large enough, this mechanism can fit well.

In order to initialize and shuffle a table synchronously, an advanced version is introduced to our mechanism to make it more efficient. The random algorithm can perfectly put a certain record i into a random location among the first i locations in the table, after moving the record previously taking up that location to location i. In normal conditions, which the records are meant to be shuffled by a column of number, especially the integers, this could be easy to represented by a function because the implementation will not change it.

There is another advantage of the random algorithm. It can be modified to adapt to certain situation that n=|T| remains unknown. When n is secret to us, the algorithm can still produce a uniform-distributed random permutation of the table. The concrete solution is as Algorithm 2.

Algorithm 2 Random shuffle algorithm

```
Initialization:
vector stack1, stack2
while amount of shuffles do
  for i = 1; i ⩽ n; i++ do
    if rand()&1 then
      stack1.pushback(array[i])
    else
      stack2.pushback(array[i])
    end if
  end for
  i = 0
  while stack1.empty()&&stack2.empty() == false do
    if stack1.empty() == true then
      array[i] = stack2.at(0)
      stack2.erase(0)
      i++, continue
    else if stack2.empty() == true then
      array[i] = stack1.at(0)
      stack1.erase(0)
      i++, continue
    end if
    if rand()&&1 then
      array[i] = stack2.at(0)
      stack2.erase(0)
      i++, continue
    else
      array[i] = stack1.at(0)
      stack1.erase(0)
      i++, continue
    end if
  end while
end while
```

4.2.3 Utility Measurement

Intuitively, there is no standard method to measure all kinds of privacy models. That's the reason why we need relative measurements. In this paper, the method we use is KL-divergence, which is short for Kullback Leibler divergence[14]. In this method, we need a benchmark first. Then we measure both of the two objects' distances to the benchmark respectively to gain the final distance between the two certain objects.

KL-divergence is not a classic distance metric on probability distributions. This is based on two certain facts. Firstly, it is not symmetric because of $D_{KL}(P,Q) \neq D_{KL}(Q,P)$. Secondly, KL-divergence does not satisfy the "tranigle inequality". But it can be used as a premetric and at the same time it generates a topology on the space of probability distributions. In real-world database, most records comply with discrete probability distributions. The distance measured by KL-divergence can be represented as follows.

$$D_{KL}(P,Q) = \sum_{i}^{x} P(i) \cdot log \frac{P(i)}{Q(i)}, \tag{5}$$

where x is the sample space. Actually, it is the expectation of the logarithmic difference between P and Q. For any i, if $P(i) = 0$, then we can say $D_{KL} = 0$, because $\lim_{x \to 0} xlogx = 0$

4.2.4 Privacy Measurement

In our model, an adversary is hard to gain useful information as we decouple the correlation between values. After processing the dataset, we need to measure the privacy level of the target dataset.

First, we discuss the scenario of one single column. "n" denotes the size of the chosen dataset. After shuffling the column, the probability of an adversary to re-identify a record is $\frac{1}{n}$. Considering we have introduce the K-means .If an adversary wants to gain all the records in a right order, one has to do this n times. We have

$$P_{col} = \{\frac{1}{n}\}^n \tag{6}$$

Then, in our model, we will shuffle M columns to ensure a certain privacy level. That means more than one column should be shuffled at most of the time. An adversary will have to re-identify all the M columns to gain the whole table. We have

$$P_{table} = \{\{\frac{1}{n}\}^n\}^M, \tag{7}$$

where M denotes the number of shuffled columns. The P_{table} is the measurement of the privacy level. The smaller the P_{table} is, the higher the privacy level will be achieved.

4.2.5 Optimization Modeling

In this paper, we want to provide a trade-off between data utility and privacy level and guarantee the efficiency at the same time. In order to get the balance, we need to calculate an optimization. We need set two thresholds. D_0 is utility threshold which means the max KL-divergence to keep data utility. P_0 denotes the threshold of privacy level, which is the max probability for an adversary to re-identify the whole table. We have

$$\begin{aligned} Minimize \quad & M \\ s.t. \quad & \\ & D_{KL} \leqslant D_0 \\ & P_{table} \leqslant P_0 \end{aligned} \tag{8}$$

Through this model, we can identify the proper M and K to control the utility and privacy level when the thresholds, D_0 and P_0, are initialized.

5. EVALUATION AND PERFORMANCE

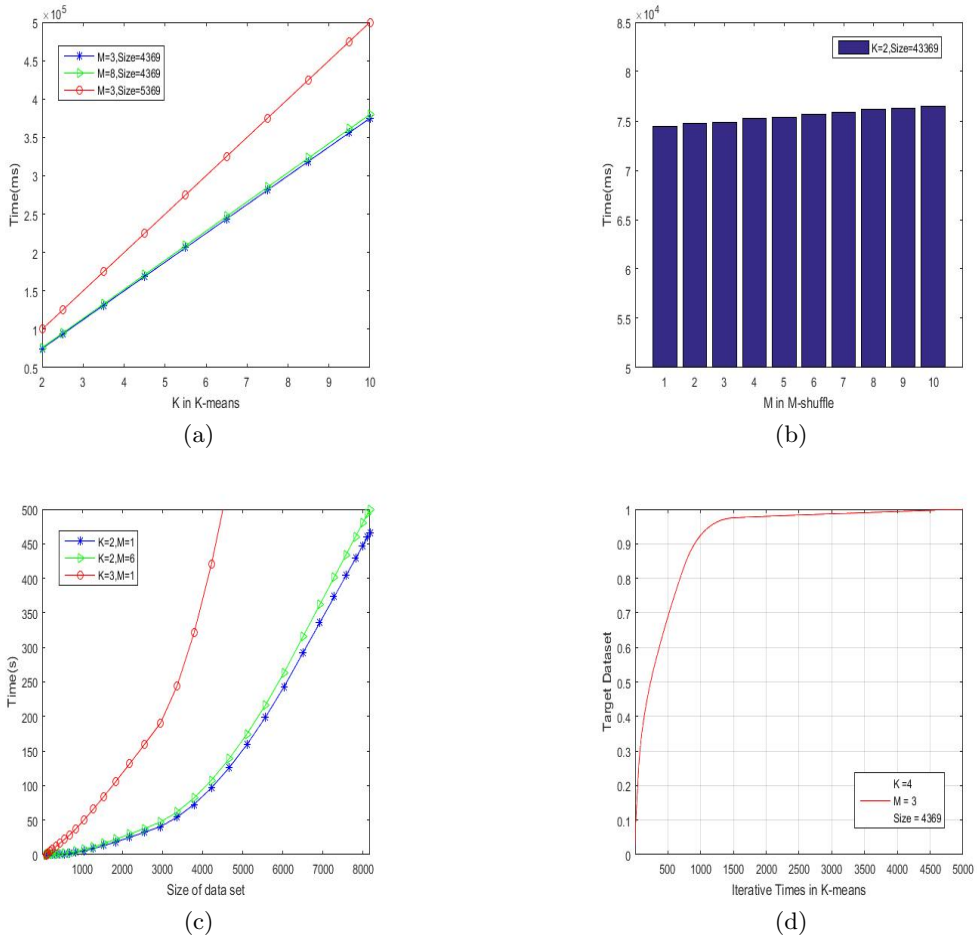


Figure 1: Time consuming with all the parameters

In this section, we evaluate the all the performances of proposed mechanism, including the efficiency and data utility. In these experiments, four kinds of privacy mechanism will be compared. We use the "Adult" data set in the experiments, which is release by the UC Irvine machine learning repository. The raw adult data are from the American census. We use six attributes from the "Adult" data set as shown in Table2. "Name" is treated as the Explicit-Identifier(EI), while the "Salary" donates the Sensitive Attribute(SA). Four other attributes fo the six are the Quasi-identifiers, which are job, age, zip code and sex. The records with missing values are removed for the accuracy of the results. There are totally records in the final version. The experiments are conducted on 64-bit windows 10 system with Core I5-4210U@2.39GHz CPU and 8G memory. The proposed algorithms will be implemented on JAVA.

5.1 Efficiency

In the first set of our experiments, we compare the running times in three situations. The first one is with the change of K, which is the number of clusters, as is shown in Figure 1a. The second one is changing with M, which means the number of QSBs, as is shown in Figure 1b. The third one fluctuates with the size of the dataset, as is shown in Figure 1c. Figure 1a and Figure 1c shows that running time follows a linear relation with K and exponential relation with size of dataset. While, it increases slightly with the increasing of M.

In Figure 1d, a convergence is found with the increase of the iteration times. After iteration times is more than 1,000, it is clear that the obtained dataset is more than 97% as same as the desirable dataset. In the experiments, we find that iteration times is always a little fewer than the size of dataset. Therefore, the iteration time is set to be 1,000 after the dataset is larger than 1,000. Figure 1c shows that running time is shorter than 10 seconds if we limit the iteration times less than 1,000. In this way, running time will not increase rapidly with the dataset size, while it will converge to the running time when the size is 1000.

5.2 Data Utility

The measurement of utility is always a big challenge to privacy researchers. As a result, we use KL-divergence to measure both the utility of the shuffled columns and of the whole table. In the series of experiments, the data size equals to 4369.

In Figure 2a, M is set to be 3. We set the original KL-distance to 1. In this figure, KL-distances of shuffled columns with $K = 1$ to $K = 6$ are compared with the original one. It is clearly shown that no matter how M changes, the

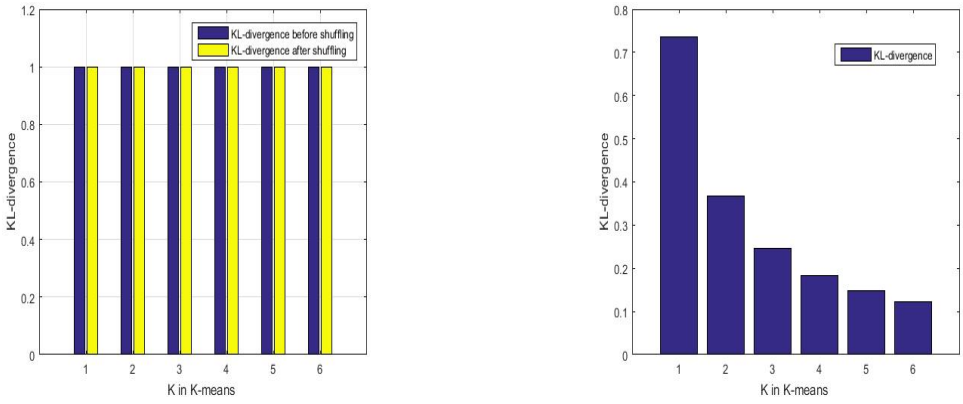


Figure 2: Information loss when k changes

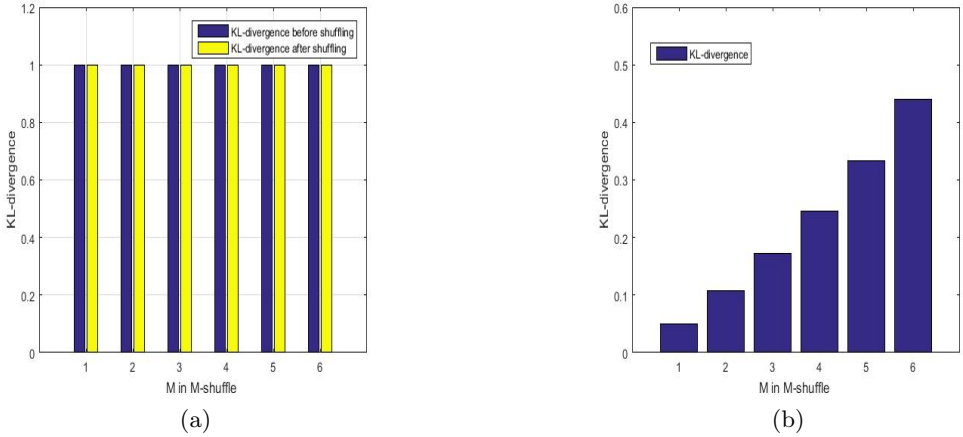


Figure 3: Information loss when m changes

KL-distances will always be a constant which equals to the original distance. That means the statistics of the shuffled columns does not alter and the utility of a certain column will keep the same after shuffling.

In Figure 2b, M is set to be 5. The KL-distance of the whole table is illustrated in this figure. We can see that with the K increases one by one, the KL-distance keeps decreasing and the decreasing rate is smaller and smaller. That shows that the KL-distance can be reduced by using a larger K. From Figure 2a and Figure 2b we can tell that introducing K-means to our model can significantly reduce the KL-distance so that the data has a better utility.

In Figure 3a, K equals to 3. We also set original KL-distance to 1 for convenience of comparing with the shuffled columns' KL-distances. Obviously, the figure shows that with the changes of M, the new KL-distance of a certain column will keep the same and equals the original one. From $M = 1$ to $M = 6$, the statistics of a single column will not change so that the utility is preserved. Our model will not harm the utility of a certain column.

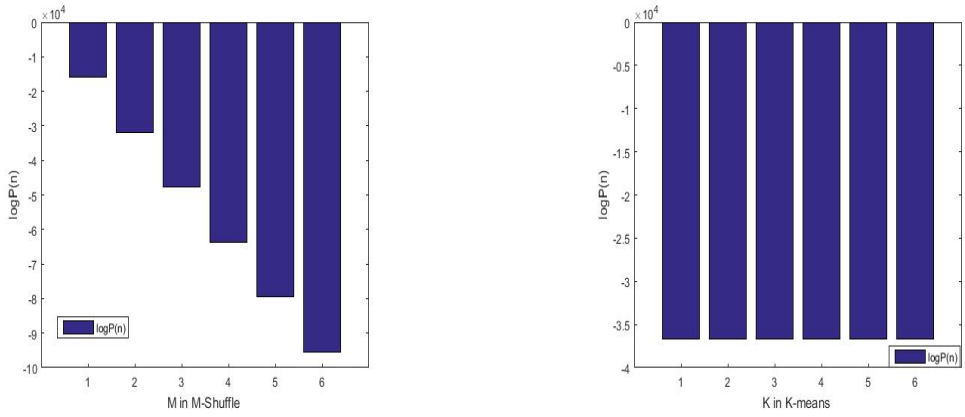
In Figure 3b, K equals to 3. This figure presents the KL-distance will increase with the change of M. When M increases, the number of shuffled columns increases accordingly. This will significantly improve the privacy level of our model. The time-consuming of an adversary will grow rapidly, which we will discuss this in next section. From Figure 3a and Figure 3b, we can tell that larger M will improve the privacy level and this only cause a slight increase of time-consuming according to Figure 1b.

5.3 Privacy Measurement

In Figure 4a, we present the possibility which the privacy of the data set would be breached by an adversary. Because of the possibilities drops too fast, we use $logP(n)$ to represent $P(n)$. It is clear that $logP(n)$ decreases at constant rate with the increase of M. That means the $P(n)$ decreases at exponential rate. Therefore, we can keep a higher privacy level by introducing a large m.

In Figure 4b, the possibility will remain the same with an increasing K. If $logP(n)$ dose not change, we can be sure that $P(n)$ does not change. K-means clustering algorithm does not lower the privacy level.

6. CONCLUSIONS AND FUTURE WORK



(a) (b)

Figure 4: $logP(n)$ when M and K changes

We present that existing models are complex and hard to implement. Therefore, we propose a more feasible and practical model using multiple shuffling. In this paper, we introduce the M-Shuffle. This model decouples the correlation among values of a column so that we can protect privacy without damaging the statistics. Experiments on real-world datasets show the effectiveness and the efficiency of the proposed model.

There are several directions for future work. The first one is that we want to carry out a better shuffle algorithm so that the model can reach a better privacy level without sacrificing utility. Then the measurement of utility is a big challenge to modern privacy study. That's why we want to propose a universe measurement by combining some modern theories, for example, the game theory, the information theory and so on.

7. REFERENCES

[1] J. Wang J. Yuan L. Xu, C. Jiang and Y. Ren. Information security in big data: Privacy and data mining. *IEEE Access*, 2:1149–1176, 2015.

[2] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 1–19, 1998.

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

[4] T. Li N. Li and S. Venkatasubramaniann. Closeness: A new privacy measure for data publishing. *IEEE Trans. Knowl. Data Eng.*, 22(7):943–956, 2010.

[5] C. Dwork. Differential privacy. *ICALP. Springer*, pages 1–12, 2006.

[6] C. Dwork, K. Nissim F. McSherry, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography, ser. TCC06*, pages 265–284. Berlin, Heidelberg: Springer-Verlag, 2006.

[7] G. Li T. Zhu, P. Xiong and W. Zhou. Correlated differential privacy: Hiding information in non-iid data set. *IEEE Transactions on Information Forensics and Security*, 10(2):229–242, 2015.

[8] Shui Yu. Big privacy: Challenges and opportunities of privacy study in the age of big data. *IEEE Access*, 4:2751–2763, June 2016.

[9] V. Marx. Biology: The big challenges of big data. *Nature*, 498:255–260, 2013.

[10] G. King. Ensuring the data-rich future of the social sciences. *Science*, 331(6018):719–721, 2011.

[11] D. Laney. 3d data management: Controlling data volume, velocity and variety. In *Gartner Report*, 2011.

[12] Z. Jorgensen, T. Yu, and G. Cormode. Conservative or liberal? personalized differential privacy. In *Procedings in 31st IEEE International Conference on Data Engineering*, pages 1023–1034. Seoul, South Korea, 2015.

[13] M.M. Pai and A. Roth. Privacy and mechanism design. *SIGecom Exchanges*, 12(1):8–29, 2013.

[14] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley & Sons, 2006.

[15] M.M. Ettefagh and Mech. Eng. Dept. Bearing fault diagnosis using hybrid genetic algorithm k-means clustering. In *Innovations in Intelligent Systems and Applications (INISTA) Proceedings*, pages 84–89. IEEE, 2014.

[16] PE Black. *Dictionary of algorithms and data structures*. 2005.