Chapter 3

Basic Definition and Problem statement

In this chapter, the basic definitions and the problem statement focused in this thesis are presented.

3.1 Basic Definition

We first present the basic definitions including dataset, *(k, e)*-Anonymous definitions, our objective function i.e. minimum summation error (MSE), and incremental data privacy breach.

Definition 1 (Dataset). Let a dataset $D = \{d_1, d_2,..., d_n\}$ be a collection of tuples that have a set of quasi-identifier attribute Q and a sensitive attribute S. D can be continuously increased with new records. The state of D at time i is denoted as D_i. A projection over the quasi identifier attributes in D_i is denoted as $D_i[Q]$. A projection over the sensitive attributes in D_i is denoted as $D_i[S]$.

Definition 2 ((k, e)-Anonymous partition). Let a set of partitions in dataset D_i be $P[D_i] = \{p_1[D_i], p_2[D_i], . . ., p_m[D_i]\}$ where the partition is $p_j[D_i] \subseteq D_i$, $\bigcup_{j=1}^{m} p_j[D_i] = D_i$, $\bigcap_{j=1}^{m} p_j[D_i] = \emptyset$ and j =1,2,...,m.

In addition, let $p_j[D_i][Q]$ and $p_j[D_i][S]$ be a projection over the quasi-identifiers attributes and the sensitive attribute in partition p_j of dataset D_i respectively.

$P[D_i]$ satisfies (k, e)-Anonymous condition when the following conditions are satisfied.

1. For all $p_j[D_i]$, $distinct(p_j[D_i][S]) \geq k$, where $distinct(p_j[D_i][S])$ is the number of distinct sensitive values in $p_j[D_i][S]$.
2. For all $p_j[D_i]$, $error(p_j[D_i][S]) \geq e$ where $error(p_j[D_i][S]) = max(p_j[D_i][S]) - min(p_j[D_i][S])$, *max* and *min* are the maximum and minimum of sensitive values from $p_j[D_i][S]$, respectively.

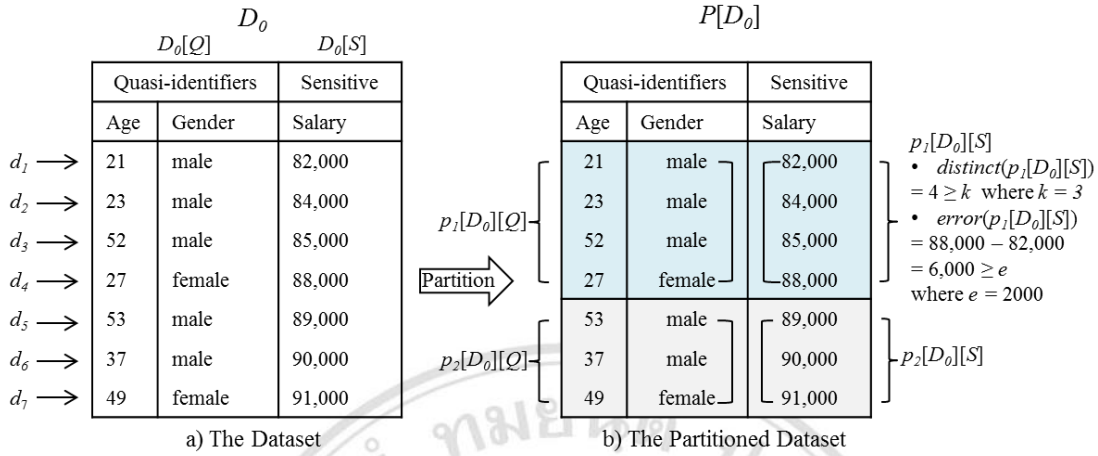

Figure 3.1: The illustration for the definition of dataset and the definition of (k, e)-Anonymous partition dataset where k = 3 and e = 2,000.

Figure 3.1 a) shows the illustration of the dataset at time 0, D_0, which has been defined in Definition 1. That is the collection of tuples from d_1 to d_7. The dataset has the attributes "Age" and "Gender" as the quasi-identifier attributes, $D_0[Q]$, and the attribute "Salary" as the sensitive attribute $D_0[S]$. The dataset is partitioned into 2 partitions, $p_1[D_0]$ and $p_2[D_0]$ as shown in Figure 3.1 b). As k is set at 3 and e is set at 2,000 in this example, both partitions are satisfied the condition of (k, e)-Anonymous partition that has been defined in Definition 2.

Not only the privacy is to be protected with the k and e parameters, but also the data utility issue must be addressed. In [10], an optimal condition for the (k, e)-Anonymous, i.e. minimizing the sum of errors, has been proposed which is as follows.

Definition 3 (Minimum Summation Error). For a dataset at time i, D_i, an error of the set of partitions $P[D_i]$ is minimized, if the summation of $error(p_j[D_i][S])$ of all of partitions is minimum.

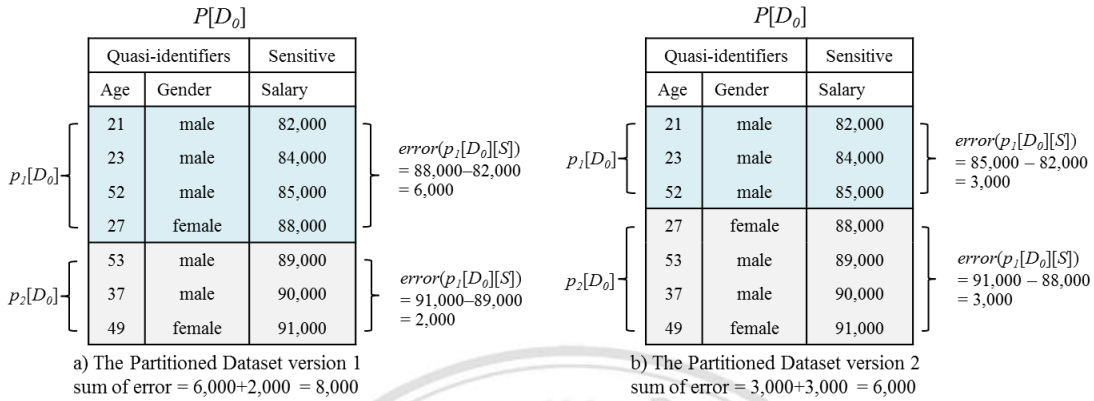

Figure 3.2: Two versions of partitioned dataset of dataset D_0

From the running example in Figure 3.1 a), there could be at least two versions of the set of partitions in dataset D_0, $P[D_0]$ as illustrated in Figure 3.2. Both versions are satisfied (*3, 2,000*)-Anonymous condition. However, the a set of partitions in dataset D_0 version 2 in Figure 3.2 b) have sum of error less than the other version in Figure 3.2 a). In fact the 2nd version of the partitioned dataset is the optimal answer to the partitioning because there is no other way to the set of partitions in dataset D_0 which is smaller than its summation error.

After the dataset is partitioned to satisfy the (*k*, *e*)-Anonymous condition, and the error is minimized, the data within each partition are shuffled as in [10].

Definition 4 (*k*, *e*)-Anonymous shuffle. Let $P'[D_i]$ be a random shuffle over a sensitive attribute S of partitioned dataset $P[D_i]$. $P[D_i] = \{p_1[D_i], p_2[D_i], \ldots, p_m[D_i]\}$. $P'[D_i] = \{p'_1[D_i], p'_2[D_i], ..., p'_m[D_i]\}$, $p'_j[D_i]$ is denoted as a set of tuples $\{d'_k \mid d'_k[Q] = d_k[Q]$ and $d'_k[S] = random(p_j[D_i][S])\}$, where the random function provides a value in $d_k[S]$ randomly without repeated value.

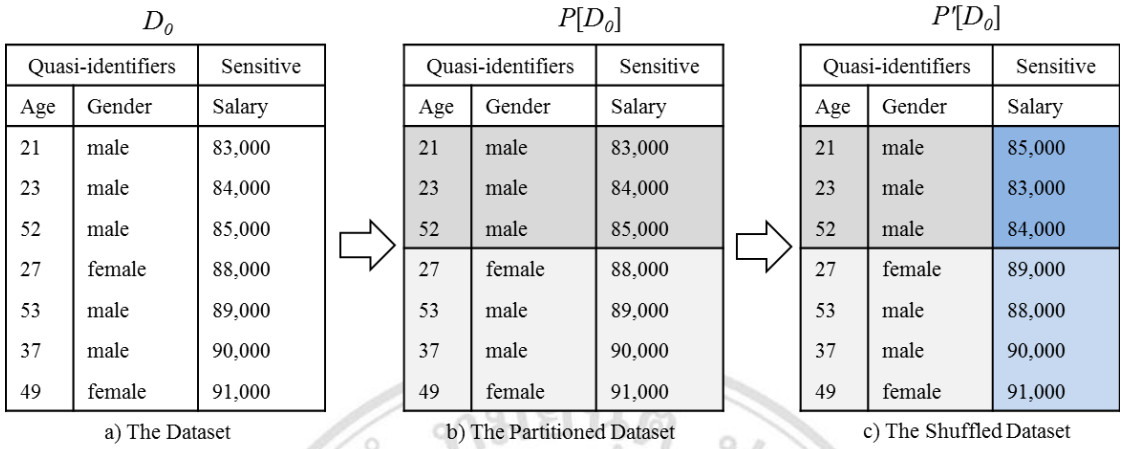

Figure 3.3: The illustration of the (*3, 2,000*)-Anonymous model

The illustration of the (*k, e*)-Anonymous model of the running example is shown in Figure 3.3. Suppose that the *k* value is set to 3 and the *e* value is set to 2000. Then, the dataset in Figure 3.3 a) can be partitioned to be $P[D_0]$ with respect to the *k* and the *e* values, as shown in Figure 3.3 b). As a matter of fact, the number of distinct sensitive values in each partition is at least 3, and the error of each partition, i.e. difference between the maximum and minimum value of each partition, is at least 2000. After the shuffling from $P[D_0]$ to be $P'[D_0]$ by *shuffle* function that has been defined in Definition 4, it can be seen that the associations between the sensitive attribute and the quasi-identifier attributes are un-linked, and thus the privacy can be preserved.

Then, we continue defining the definitions for a kind of privacy breach when additional data are appended as follows.

Definition 5 (Incremental Privacy Breach). Given any two the sets of partitions in dataset $P'[D_i]$, and $P'[D_j]$ where $j > i$. The (*k, e*)-Anonymous incremental privacy breach from $P'[D_j]$ to $P'[D_i]$ is denoted as $P'[D_j] \rightarrow_{(k, e)} P'[D_i]$. Such a breach can be categorized into two cases as follows.

Difference privacy breach. For any partition $p'_a[D_i]$ and $p'_b[D_j]$, let p^-_{ab} be a pair of $p^-_{ab}[Q]$ and $p^-_{ab}[S]$ where $p^-_{ab}[Q] = p'_a[D_i][Q] - p'_b[D_j][Q]$, and $p^-_{ab}[S] = p'_a[D_i][S] - p'_b[D_j][S]$.

a. $P'[D_j] \rightarrow_{(k,\,e)} P'[D_i]$, *if*($\exists p'_a[D_i]$ and $\exists p'_b[D_j]$ in which $|p^{-}_{ab}[Q]| > 0$) $\wedge$ (*distinct*($p^{-}_{ab}[S]$) < $k \vee$ *error*($p^{-}_{ab}[S]$) < e).

b. $P'[D_j] \rightarrow_{(k,\,e)} P'[D_i]$, *if*($\exists p'_a[D_i]$ and $\exists p'_b[D_j]$ in which $|p^{-}_{ba}[Q]| > 0$) $\wedge$ (*distinct*($p^{-}_{ba}[S]$) < $k \vee$ *error*($p^{-}_{ba}[S]$) < e).

Note that $\wedge$ represents the conjunction operator, and $\vee$ represents the disjunction.

Intersection privacy breach. Let $p^{\cap}_{ab}$ be a pair of $p^{\cap}_{ab}[Q]$ and $p^{\cap}_{ab}[S]$ where $p^{\cap}_{ab}[Q] = p'_a[D_i][Q] \cap p'_b[D_j][Q]$, and $p^{\cap}_{ab}[S] = p'_a[D_i][S] \cap p'_b[D_j][S]$. $P'[D_j] \rightarrow_{(k,e)} P'[D_i]$, *if*($\exists p'_a[D_i] \wedge \exists p'_b[D_j]$ in which $|p^{\cap}_{ab}[Q]| > 0$) $\wedge$ (*distinct*($p^{\cap}_{ab}[S]$) < $k \vee$ *error*($p^{\cap}_{ab}[S]$) < e).

$P'[D_0]$

	Quasi-identifiers		Sensitive
	Age	Gender	Salary
$p'_a[D_0]$	21	male	85,000
	23	male	82,000
	52	male	84,000
$p'_b[D_0]$	27	female	89,000
	53	male	90,000
	37	male	91,000
	49	female	88,000

a) The Shuffled Dataset at time 0

$P'[D_1]$

	Quasi-identifiers		Sensitive
	Age	Gender	Salary
$p'_c[D_1]$	29	female	81,000
	30	female	80,000
	21	male	82,000
$p'_d[D_1]$	23	female	84,000
	23	male	85,000
	52	male	83,000
$p'_e[D_1]$	27	female	91,000
	53	male	88,000
	37	male	89,000
	49	female	90,000

b) The Shuffled Dataset at time 1

Figure 3.4: The shuffled dataset at time 0 and 1 that are satisfied (*3, 2000*)-Anonymous which minimum summation error

Before further discussion on the incremental privacy breach issues will be presented, let us consider the example in Figure 3.4 for the understanding of the incremental privacy breach. Figure 3.4 a) illustrates $P'[D_0]$, that is the shuffled dataset at time 0, and Figure 3.4 b) illustrates $P'[D_1]$, that is the shuffled dataset at time 1. Note that $P'[D_1]$ is the result of adding 3 new tuples to D_0. It can be seen that both shuffled

datasets satisfied (*3*, *2000*)-Anonymous condition, also, they are the optimal solution that is subjected to the summation errors. Next, let us illustrate the incremental privacy breach for this example.

A method to break the privacy protection based on (*k*, *e*)-Anonymous model, when considering $P'[D_0]$ together with $P'[D_1]$, $P'[D_0] \rightarrow_{(3,\ 2000)} P'[D_1]$, is to determine the difference and intersection privacy breach between each partition in $P'[D_0]$ and each partitions in $P'[D1]$, one by one. According to the difference and intersection privacy breach definitions in Definition 5, the set of pair of partitions to find out an incremental privacy breach in the example in Figure 3.4 is $\{(p'_a[D_0], p'_c[D_1]), (p'_a[D_0], p'_d[D_1]), (p'_a[D_0], p'_e[D_1]), (p'_b[D_0], p'_c[D_1]), (p'_b[D_0], p'_d[D_1]), (p'_b[D_0], p'_e[D_1])\}$. If at least one of them has an incremental privacy breach then it can conclude this 2 version of datasets have an incremental privacy breach from $P'[D_0]$ to $P'[D_1]$, $P'[D_0] \rightarrow_{(3,\ 2000)} P'[D_1]$.

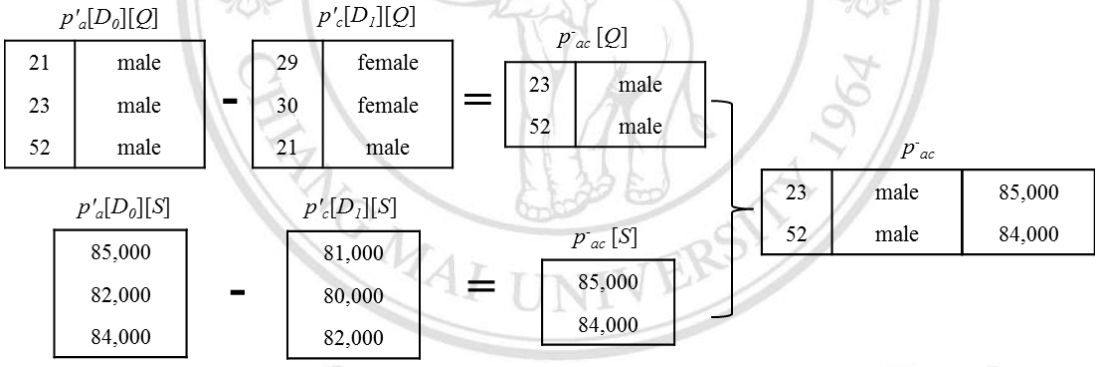

Figure 3.5: Difference privacy breach from $p'_a[D_0]$ to $p'_c[D_1]$, p^-_{ac}

An example of such difference privacy breach from $p'_a[D_0]$ to $p'_c[D_1]$ shows in Figure 3.5. The illustration starts by difference calculation $p^-_{ac}[Q]$ between $p'_a[D_0][Q]$ and $p'_c[D_1][Q]$ and it calculates $p^-_{ac}[S]$ from the difference of $p'_a[D_0][S]$ and $p'_c[D_1][S]$. Then, $p^-_{ac}[Q]$ and $p^-_{ac}[S]$ are composed to p^-_{ac}. The result of composing, p^-_{ac}, are the 2 tuples indicating (*3*, *2000*)-Anonymous condition is not satisfied. So, this can conclude $P'[D_0] \rightarrow_{(3,\ 2000)} P'[D_1]$.

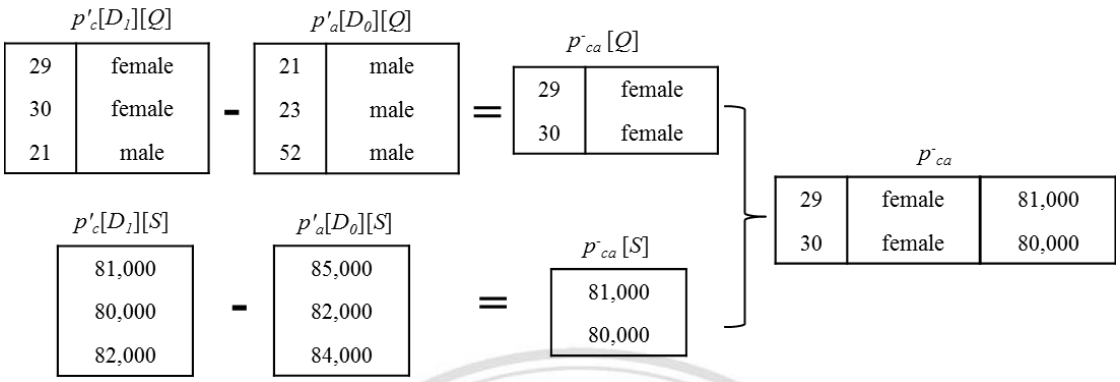

Figure 3.6: Difference privacy breach from p'_c $[D_1]$ to $p'_a[D_0]$, p^-_{ca}

The other way to determine a difference privacy breach by reverse different is illustrated in Figure 3.6. That is the difference privacy breach from $p'_c[D_1]$ to $p'_a[D_0]$. The illustration starts by calculating p^-_{ca} $[Q]$ from the difference between $p'_c[D_1][Q]$ and $p'_a[D_0][Q]$, and calculating of $p^-_{ca}[S]$ from the difference between $p'_c[D_1][S]$ and $p'_a[D_0][S]$. Then, $p^-_{ca}[Q]$ and $p^-_{ca}[S]$ are composed to p^-_{ca}. The result is, p^-_{ca}, which composed of the 2 tuples that are not satisfied (*3, 2000*)-Anonymous condition. So, it can be concluded that $P'[D_0] \rightarrow_{(3,\ 2000)} P'[D_1]$.

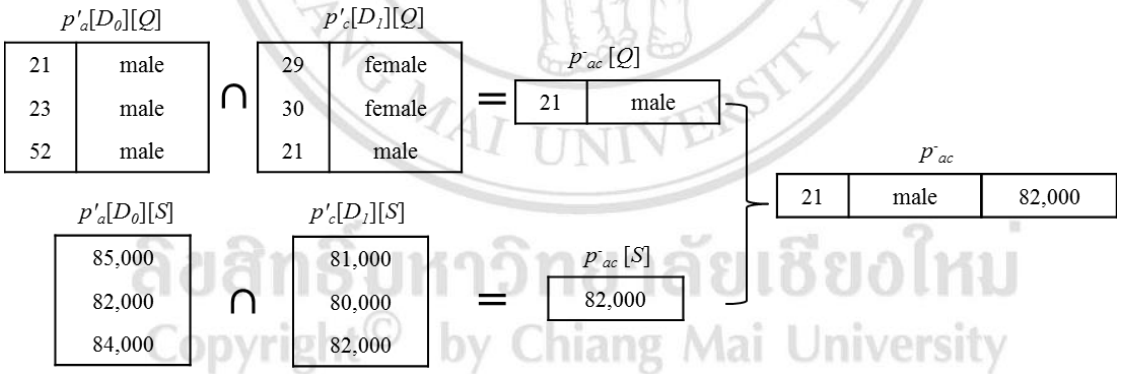

Figure 3.7: Intersection privacy breach between $p'_a[D_0]$ and p'_c $[D_1]$, $p^{\cap}_{ac}$

Last, Figure 3.7 illustrates an intersection privacy breach between $p'_a[D_0]$ and $p'_c[D_1]$. From the calculation of $p^{\cap}_{ac}[Q]$ from $p'_a[D_0][Q]$ intersecting with $p'_c[D_1][Q]$ and calculation of $p^{\cap}_{ac}[S]$ from $p'_a[D_0][S]$ intersecting with $p'_c[D_1][S]$, we can determine $p^{\cap}_{ac}$ from $p^{\cap}_{ac}[Q]$ and $p^{\cap}_{ac}[S]$. It shows that $p^{\cap}_{ac}$ has only 1 tuple that is not satisfied (*3, 2000*)-Anonymous condition. So, it can conclude that $P'[D_0] \rightarrow_{(3,\ 2000)} P'[D_1]$.

It can be seen from the examples that the incremental privacy breaches occur because of the difference and the intersection conditions. The new partitions p^{-}_{ac}, $p^{\cap}_{ac}$, and p^{-}_{ca} in the examples lead to an incremental privacy breach to the previous dataset in terms of the two types of condition: the number of distinct values and the errors bound. Furthermore, such a problem can be escalated when there are multiple versions of datasets and not only two versions.

3.2 Problem Definition

Now, we are ready to define our problem to be addressed in this thesis as follow.

Given a set of released datasets $P = \{P'[D_0], P'[D_1], \ldots, P'[D_{n-1}]\}$, the current dataset D_n to be released, the value of k, and the value of e. The incremental (k, e)-Anonymous privacy preservation problem is to determine the $P'[D_n]$ that satisfies (k, e)-Anonymous condition as well as to prevent the incremental privacy breach against all of the previous released datasets, and the sum of the errors of $P'[D_n]$ is minimized.

3.3 Existing Issues

From the problem definition in the previous section, it can be seen that there are a few issues aside from the privacy preservation in order to solve such problem.

3.3.1 Efficiency issue

An approach to solve the problem is brute force process to find the optimal partitioned dataset that is subjected to the summation errors. The illustration of brute force process is shown in Figure 3.8. The brute force could starts by partitioning the current dataset to find all possible partitioned dataset. The complexity of the partitioning process is $O(2^n)$ where n is the number of tuples of current dataset. Next, each version of partitioned dataset are examined for (k, e)-Anonymous conditions, and an incremental privacy breach conditions with all previously released datasets. After the examination, it leaves some partitioned datasets that have no breach. Last, the dataset with minimum sum of error is to be chosen as the solution. It can be seen that, only partitioning process in the brute force has in fact, exponential complexity. So, the brute

force process might not be appropriate to use in the real world scenarios where the size of the dataset is rather large.

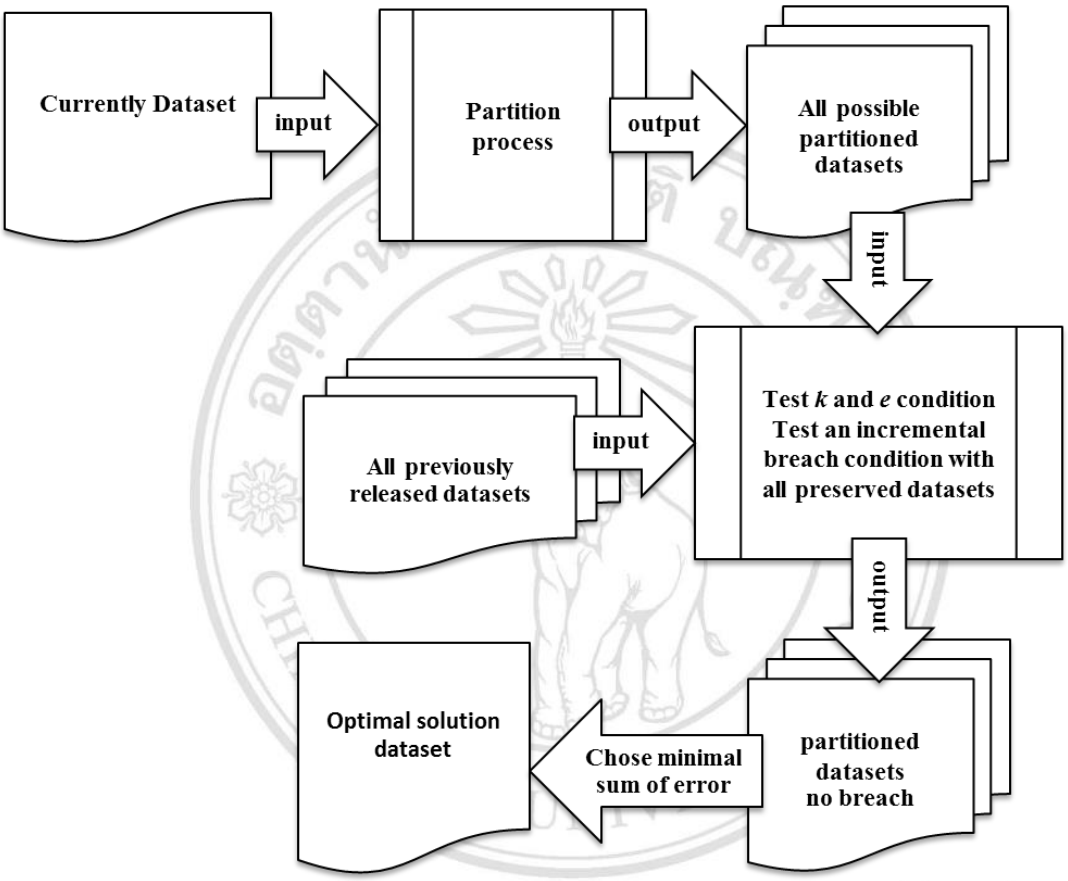

Figure 3.8: Brute force of incremental process

The other solution is a naïve re-applying algorithm that re-applies the existing static algorithm proposed in [10], the minimum summation of error algorithm which has been already discussed in Chapter 2. We present it here again in Figure 3.9 for the sake of illustration.

Input:
{ $P'[D_0],\ldots, P'[D_{n-1}]$}: a set of all previously released datasets
D_n: a dataset that t_n, sorted by sensitive values
k: a threshold for the minimum number of distinct values
e: a minimum *error* of the threshold values
Output:
$P'[D_n]$: the portioned dataset, that does not have an incremental privacy breach with $P'[D_0],\ldots, P'[D_{n-1}]$, and it has a minimum sum of *error*
partition: the partition information
error: the *error* information
Method:

```
1  error[0] = 0
2  partition[0] = 0
3  for i =1 to Dn.size
4     error[i] = infinity
5     partition[i] = partition[i - 1]
6     for j = 1 to i
7        if {di[Dn][S],…, dj[Dn][S]} satisfy k and e
8          if {di[Dn][S],…, dj[Dn][S]} doesn't has an incremental breach with P'[Dn-1]
9            current_error = di[Dn][S] – dj[Dn][S]
10         else
11           current_error = infinity
12         end if
13       else
14         current_error = infinity
15       end if
16       temp = error[j - 1] + current_error
17       if temp < error[i]
18         error[i] = temp
19         partition[i] = j
20       end if
21     end for
22 end for
```

Figure 3.9: The Naïve Re-Applying Algorithm

From Figure 3.9, the idea of the algorithm is as follows. First, for each considered partition of the dataset to be released D_n, the (k, e)-Anonymous condition is evaluated. If it satisfies the condition, then the partition is considered for any incremental privacy breach occurs against all of the released datasets. If such a partition has an incremental privacy breach with at least one of the released datasets, then the current partition can be discarded. For the incremental privacy breach determination, the

algorithm must compute the difference and the intersection results between the current considered partition and each partition for each released dataset. For simplicity, suppose that the number of tuples in each released dataset is n, and the number of released datasets is m. The complexity of this computing as well as the whole algorithm is $O(mn^3)$. However, a naïve re-applying algorithm is not proper to use in a long run. Because, when the number of released datasets, m, is increased, the algorithm will be very inefficient.

3.3.2 Effectiveness issue

However, it is not trivial to re-apply a naïve algorithm by considering only one previous released dataset for efficiency issue. As the solution dataset has also to be optimal subjected to the summation of error. And the solution dataset must not have an incremental privacy breach with all previously released datasets. Nevertheless, the re-applying a naïve algorithm by considering only one previous released dataset may give the solution dataset that have an incremental privacy breach from solution dataset to another previously released dataset.

$P'[D_0]$

	Quasi-identifiers		Sensitive
	Age	Gender	Salary
$p'_a[D_0]$	21	male	85,000
	23	male	82,000
	52	male	84,000
$p'_b[D_0]$	27	female	89,000
	53	male	90,000
	37	male	91,000
	49	female	88,000

a) The Shuffled Dataset at time 0

$P'[D_1]$

	Quasi-identifiers		Sensitive
	Age	Gender	Salary
$p'_c[D_1]$	29	female	81,000
	30	female	80,000
	21	male	82,000
	23	female	84,000
	23	male	85,000
	52	male	83,000
$p'_d[D_1]$	27	female	91,000
	53	male	88,000
	37	male	89,000
	49	female	90,000

b) The Shuffled Dataset at time 1

$P'[D_2]$

	Quasi-identifiers		Sensitive
	Age	Gender	Salary
$p'_e[D_2]$	29	female	81,000
	30	female	80,000
	21	male	82,000
$p'_f[D_2]$	23	female	84,000
	23	male	85,000
	52	male	83,000
$p'_g[D_2]$	27	female	91,000
	53	male	88,000
	37	male	89,000
	49	female	90,000
$p'_h[D_2]$	40	male	96,000
	50	male	95,000
	32	female	94,000

c) The Shuffled Dataset at time 2

Figure 3.10: 3 versions of the solution dataset from naïve re-applying algorithm by considering one version of previously released dataset

Quasi-identifiers		Sensitive
Age	Gender	Salary
21	male	82,000

a) Result of intersection between p'_a and p'_e

Quasi-identifiers		Sensitive
Age	Gender	Salary
23	male	85,000
52	male	84,000

b) Result of difference between p'_a and p'_e

Quasi-identifiers		Sensitive
Age	Gender	Salary
29	female	81,000
30	female	80,000

c) Result of difference between p'_e and p'_a

Quasi-identifiers		Sensitive
Age	Gender	Salary
23	male	85,000
52	male	84,000

d) Result of intersection between p'_a and p'_f

Quasi-identifiers		Sensitive
Age	Gender	Salary
21	male	82,000

e) Result of difference between p'_a and p'_f

Quasi-identifiers		Sensitive
Age	Gender	Salary
23	female	83,000

f) Result of difference between p'_f and p'_a

Figure 3.11: Results of an intersection and a difference between a partition in D_0 and a partition in D_2

The Figure 3.10 illustrates such scenario where an incremental privacy breach occurs from using a naïve re-applying algorithm by considering only single version of previously released dataset. All solution datasets in this figure are satisfied (*3*, *2000*)-Anonymous conditions. The algorithm generates the solution dataset, $P'[D_1]$, by considering $P'[D_0]$. It can be seen, the solution dataset $P'[D_1]$ does not have an incremental privacy breach with $P'[D_0]$, i.e. $p^-_{ac,}$ p^-_{ca}, and $p^{\cap}_{ac}$ satisfy (*3*, *2000*)-Anonymous conditions. Now consider the solution dataset $P'[D_2]$, the algorithm generates $P'[D_2]$ by considering only $P'[D_1]$. It can be seen that, the solution dataset $P'[D_2]$ does not have an incremental privacy breach with $P'[D_1]$, i.e. $p^-_{ce,}$ p^-_{ec}, $p^{\cap}_{ce}$, $p^-_{cf,}$ p^-_{fc}, and $p^{\cap}_{cf}$ satisfy (*3*, *2000*)-Anonymous conditions. However, an incremental privacy breach is occurred between $P'[D_2]$ and $P'[D_0]$ or $P'[D_2] \rightarrow_{(3,\ 2000)} P'[D_0]$, $p^-_{ae,}$ p^-_{ea}, $p^{\cap}_{ae}$, $p^-_{af,}$ p^-_{fa}, and $p^{\cap}_{af}$ do not satisfy (*3*, *2000*)-Anonymous conditions as shown in the Figure 3.11.

3.4 Summary

This thesis aims to develop an algorithm that considers only one version of previously released dataset and the current dataset as the inputs for finding an optimal solution dataset for releasing. Such work will be presented in the next chapter.