Achieving k-anonymity using Full Domain Generalization

Amit Kumar Pal

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

May 2014

Achieving k-anonymity using Full Domain Generalization

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

(Specialization: Information Security)

by

Amit Kumar Pal

(Roll No.- 212CS2366)

under the supervision of

Prof. Korra Sathya Babu

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India

May 2014

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Certificate

This is to certify that the work in the thesis entitled ***Achieving k-anonymity using Full Domain Generalization*** by ***Amit Kumar Pal*** is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology with the specialization of Information Security in the department of Computer Science and Engineering, National Institute of Technology Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Place: NIT Rourkela
Date: June 2, 2014

(Prof. Korra Sathya Babu)
Professor, CSE Department
NIT Rourkela, Odisha

Acknowledgment

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof Korra Sathya Babu who has been the guiding force behind this work. I want to thank him for introducing me to the field of Privacy Preserving Data Publishing and giving me the opportunity to work under him. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his invaluable advice and assistance it would not have been possible for me to complete this thesis. I am greatly indebted to her for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

Amit Kumar Pal

Roll-212cs2366

Abstract

Preserving privacy while publishing data has emerged as key research area in data security and has become a primary issue in publishing person_specific sensitive information. How to preserve one's privacy efficiently is a critical issue while publishing data.

K-anonymity is a key technique for de-identifying the sensitive datasets. In our work, we have described a framework to implement most of the k-anonymity algorithms and also proposed a novel scheme that produces better results with real-world datasets. Additionally, we suggest a new approach that attains better results by applying a novel approach and exploiting various characteristic of our suggested framework. The proposed approach uses the concept of breadth-search algorithm to generalize the lattice in bottom-up manner. the proposed algorithm generates the paths using predictive tagging of the nodes in the lattice in vertically.the proposed algorithm has less execution time than other full domain generalization algorithms for k-anonymization.

Keywords: k-anonymity, Data Privacy, domain generalization,Quasi-Identifier, data utility etc.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over last 20 years, the digitization of our daily lives has led to an increase in the data collected by individuals, corporations, and governments. This digitally available data (known as microdata) has created a good opportunity for decision making based on available information. Because of mutual benefits, or by organization's policies, publication of digitally available data is required to improve decision making. But the collected microdata in its native form may contain person specific sensitive information of individuals whose privacy can be violated if the original data is published.

So the important task is to protect the privacy of this microdata. There exists some guidelines, agreements and policies about how and what data should be published so that the data remains useful for research and analysis and at the same, individual's privacy is preserved, referred as privacy preserving data publishing (PPDP).

In this thesis, we consider only preserving of information privacy, which protects sensitive information from being brought to the attention of others. Privacy preserving is the ability to limit the diffusion and use of one's personal data. Privacy can refer to an individual where nobody should know about any entity after performing data mining or an organization to protect knowledge about a collection of entities. Various approaches followed for individual privacy preserving are data obfuscation, value swapping, perturbation, etc. Each organization adopts a framework for disclosing individual entity values to the public.

1.1 Privacy-Preserving Data Publishing

Privacy preserving data publishing($PPDP$) is an approach to publish practically useful data without violating individuals privacy. PPDP focuses on data anonymization that attempt to conceal the identity of record holders, considering that private data must be maintained for data analysis [2]. PPDP consist of two phases: Data collection and data publication.

1. Data collection: in this phase, the original data from record holders is retrieved by the data publisher.

2. Data publishing: in this phase, the data retrieved by record holders in data collection phase, is released to data recipient for analysis and mining purpose. A real time scenario of PPDP is given as follows:

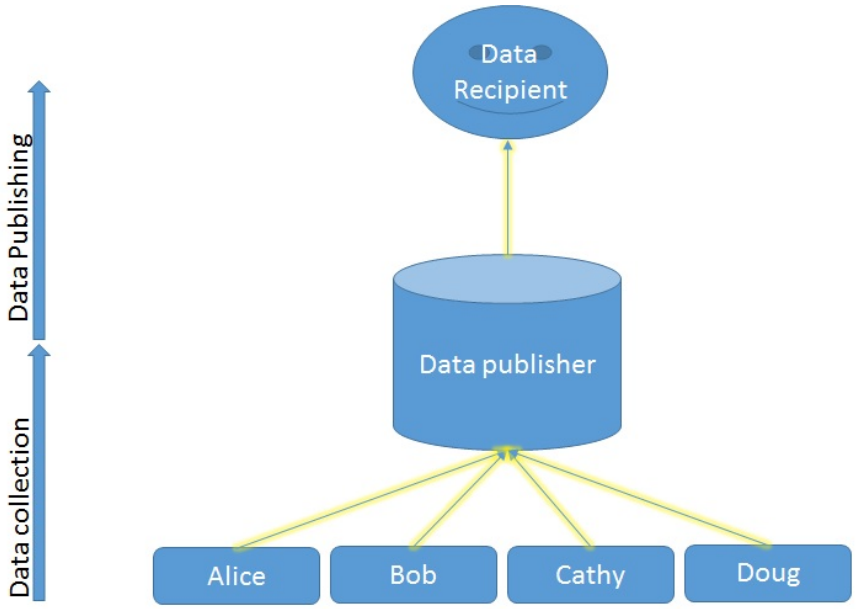

Figure 1.1: Privacy Preserving Data Publishing [2]

In this example, we can compare this with the hospital patient scenario where Alice, Bob, Cathy, Doug are the patient (Record holders) and data publisher (hospital) collects the information from record owners and gives it medical center (Data recipient) for research and analysis purpose. Finally data recipient perform data mining to retrieve useful information. On the basis of trust level the data

publisher is categorizes in two models: trusted data publisher and untrusted data publisher.

1: Trusted Data publisher: In this model, the record holders know that the data publisher is reliable and they are willing to provide their personal information for analysis [13].

2: Untrusted data publisher: In it, the publisher may not be reliable and may try to gain confidential information from the record holders.

1.2 Anonymization Approach

In basic scenario of privacy preserving data publishing, the published data table has the following form:

D (*Explicit_Identifier, Quasi-Identifier,Sensitive_Attributes, Non_ Sensitive_Attributes*)

Where

Explicit_Identifier: it is a group of attributes (for e.g. voter_id, Name etc.), able to identify individual record explicitly.

Quasi-Identifier: A group of attributes from a table whose combination can be used to identify some other record from dataset. Quasi-identifers may be used to re-identify an individual record from the table.For example [2] combination of (Job ,Postcode,date_of_birth) of all these attribute may used to determine any individual record from the table, to his/her medical problem.

Sensitive-attributes:

Sensitive Attributes contain the sensitive person-specific information which an Individual will never want to disclose it. Non-Sensitive attibutes are those who do not come under remaining three types of attributes.

1.3 *k*-anonymity:

In the generalized table, a tuple must be indistinguishable from (k-1) other tuples having the same quasiIdentifier. A relation is consist of quasiidentifier and non-quasiidentifier attributes in which quasiIdentifier attributes needs to be anonymized

Table 1.1: Original Table

Job	**Birth**	**Zipcode**	**Disease**
Engineer	1970	9008	Hepatitis
Engineer	1960	9008	Hepatitis
Engineer	1960	9005	HIV
Engineer	1960	9006	HIV
lawer	1970	9008	HIV
lawer	1970	9008	Flue

because their combination can reidentify the individual's record. Consider t is a tuple in the generalized table, the value of the i_{th} tuple is t_i[C].

A relation, T_1 satisfies k-anonymity if for each tuple $t_{i0} \in T_1$,there are (k-1) other tuples $t_{i1}, t_{i2}, t_{i3}, ..., t_{i(k-1)} \in T_1$ such that $t_{i0}[C] = t_{i1}[C] = t_{i2}[C] = t_{i3}[C] = ... = t_{i(k-1)}[C]$

1.4 Anonymization

Protection of individual's confidential data is of prime importance. Releasing individual's data (containing sensitive information) publicly might cause risk for individual's privacy [5]. so the first step to anonymize the table is to remove the explicit identifier because this attribute directly reveals identity of record holder.

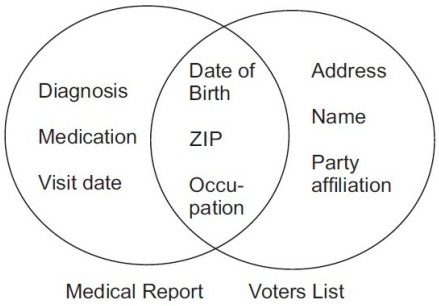

Figure 1.2: Linking attack to identify record holder

But L seweney's survey [6] shows that removing explicit identifier is not enough to protect individual's privacy. the survey shows that approximately 87 percentage of USA citizens can be re-identified with the help of birth_data, zipcode and gen-

der attributes when linked with the voter list database to the published medical database. According to this survey, the record holder is linked with the publicly available databases and re-identified with the help of quasi-identifiers(date of birth, gender and age).

for this linking attack, [7]adversary requires only these two prior knowledge: the record of the victim should be present in the published database and the quasi-identifier of the victim.

Table 1.2: 2-anonymous table

Job	**Birth**	**Zipcode**	**Disease**
Engineer	*	9008	Hepatitis
Engineer	*	9008	Hepatitis
Engineer	1960	900*	HIV
Engineer	1960	900*	HIV
lawer	1970	9008	HIV
lawer	1970	9008	Flue

To protect from this linking attack, the data table must be anonymized to the following form:

T(QID' , Sensitive-Attributes, Non-Sensitive_Attributes),

where QID' is ananonymized type of the given QID generated by performing anonymization actions to the attributes in QID in the original data table D. Anonymization approach conceal the information of some quasi-identifier so that few other records also become similar to that record in the published table. now with the generalized table,if an adversary links a record holder to a tuple in QID', the record holder is also matched with *(k-1)* other records in the QID'.

The main goal of the anonymization task is to generate an anonymous table T that fulfills the basic guidelines of a given privacy model and also contains as much useful information as possible. To estimate the utility of the anonymous data, there are some metrics like general purpose, special purpose, trade-off metrics.

1.5 Attack Models in Privacy Preserving and Data Publishing

According to Dalenius [1977] [8], the privacy protection is not allowing an adversary to gain any person-specific sensitive information of a targeted individual even though he has some background knowledge from external sources. The attack models in the PPDP can be categorized in two ways based on their attack principles: [9]In first type, if an adversary finds a way to map a record holder to a tuple present in the published anonymized table or to an sensitive attribute in the table. these are known as linking attacks.

In second type, main focus of the adversary is to gain information about the victim with the help of previously known knowledge (background knowledge).

1.5.1 Record Linkage

Record linkage refers to the mapping of some records to the targeted victim in the publicly released table based on quasi-identifier of the victim. If the victim's quasi-identifier matches with the records in the released table then the adversary faces less no. of possibilities for targeted record.With some additional information

From given tables 1.3 to 1.6, The research center maps the records in table 1.3 and 1.4 based on same quasi-identifiers present in both table it gain sensitive information , here by joining these two tables 1.3 and 1.4 for quasi -identifier job, sex and age it can found that male whose age is 38 and profession is lawyer suffers from HIV is mapped to Doug.

To avoid such type of attack by record linkage , a new technique is proposed by Sweeney ,Samrati [9] in this model for each set of all quasi-identifiers having same value in table must have atleast k number of records .The benefit of this model is that there are other (k-1) tuples that are mapped to same quasi-identifier set with probability of attack 1/k. As it shown in table 1 for quasi-identifier (job,birth,postcode).

Subset Property of k-anonymity

If a table is k anonymous with a set of quasi-identifiers Q , then the must satisfy

k anonymity with respect to all subset Q [10].

Table 1.3: Patient Table

Job	**Sex**	**Age**	**Diease**
Engineer	Male	35	Hepatitis
Engineer	Male	38	Hepatitis
Lawyer	Male	38	HIV
Writer	Female	30	Flu
Writer	Female	30	HIV
Dancer	Female	30	HIV
Dancer	Female	30	HIV

Table 1.4: 3-Anonymous Table

Job	**Sex**	**Age**	**Desiese**
Professional	Male	35-40	Hepatitis
Professional	Male	35-40	Hepatitis
Professional	Male	35-40	HIV
Artist	Female	30-35	Flu
Artist	Female	30-35	HIV
Artist	Female	30-35	HIV
Artist	Female	30-35	HIV

Table 1.5: External Table

Name	**Job**	**Sex**	**Age**
Alice	Writer	Female	30
Bob	Engineer	Male	35
Cathy	Writer	Female	30
Doug	Lawyer	Male	38
Emily	Dancer	Female	30
Fred	Engineer	Male	38
Gladys	Dancer	Female	30
Henry	Lawyer	Male	30
Irene	Dancer	Female	32

Table 1.6: 4 Anonymous External Table

Name	**Job**	**Sex**	**Age**
Alice	Artist	Female	[30-35)
Bob	Professional	Male	[35-40)
Cathy	Artist	Female	[30-35)
Doug	Professional	Male	[35-40)
Emily	Artist	Female	[30-35)
Fred	Professional	Male	[35-40)
Gladys	Artist	Female	[30-35)
Henry	Professional	Male	[30-35)
Irene	Artist	Female	[30-35)

(X,Y)-Anonymity The assumption of k anonymity [11] is that each records present in anonymized table is unique existence in real life which may not be true for example let a patient may have more than one disease at a time so it might be possible it its quasi-identifier present in original table may satisfy k but in reality their records links to single identity. [12]To avoid this problem [28] proposed (X, Y)-anonymity, where X and Y are disjoint sets of attributes. A_Y (X) is the anonymity for set of quasi-identifiers X .it is the total number of unique Y values with respect to same X. So the table satisfy (X,Y) anonymity if A_Y (X) $\geq$ K.

It states that for set of attribute size(quai-identifier) X must be mapped to at least Y unique values. [13]Eg. as in previous case ,X is set of {Job,age,gender} and Y is the sensitive_attribute so for each same set of X there must be at least Y different values.

1.5.2 Attribute Linkage

In this attack , attacker gain some information about his sensitive attribute from the released table , even though attacker is not able to link the victim with any individual published record . [14]From the table 1.6, attacker can find that all the female having age 30 whose profession is dance suffer from HIV.so {Dance,Female ,30} is confidence 100 percent HIV by this information it found that Emily suffers from HIV. L -Diversity. To prevent from attribute linkage attack it is purposed by Machanavjjhala [13] [14] .Its necessary conditions is every equivalence of released table must have at least l different values.The fundamental concept is to avoid attribute linkage as we seen from the last example if there will be different unique sensitive values it prevents attribute linkage. But probabilistic attacks can not be avoided by this because flu is very common disease compared to HIV. The released table satisfy l-diverse property if for all qid group

$$-\sum P(qid,s)log(P(qid,s)) \geq log(l) \tag{1.1}$$

Here S is sensitive_attribute, P(qid,s) is a part of records whose sensitive value is s for the total records whose equivalence class is group denoted by qid [15]. The more uniformly distributed sensitive values in each equivalence class group qid higher will be the entropy of sensitive attribute. So higher value of entropy in the released table , lesser is the chances probabilistic attack, higher value of threshold l increases its privacy and lesser is the information gain by attacker from released table.

Limitations

The major drawback of entropy l -diversity is it is not able to the measure of probabilistic attack [16] for eg as it is calculated entropy is 1.8 but in second equivalence group out of 4 records 3 suffers from HIV from table 1.6, which is easy for probabilistic attack.

1.6 Anonymization Operations

The table which contains the original records values of each individual person do not provide any privacy. To publish it and to preserve the privacy of each individual person, some operations have to be performed . Anonymization is a technique to solve the problem of data publishing, it while keep the sensitive information of record owner which is to be used for data analysis it hides the explicit identity of that record owner from the table which is going to be published.

Anonymization can be done by using following operations [17]

1. Generalization
2. Suppresion

1.6.1 Generalization

Generalization modifies the quasi-identifier original most specific value to the some generalized values of specific description, eg specific form date of birth to generalized to year only while hiding month and date value. Full-domain generalization scheme [6] while generalizing, for all records and for any quasi-identifier, generalization is applied upto few level of hierarchy tree For eg. If a equivalence class of {writer, dancer } is generalized to Artist then other equivalence of {Engineer ,Lawyer } must be generalized to Professional. Generaized table is consistent and it is used in Global recoding algorithms, but the major drawback of this is data loss is very high.

.

1. Subtree Generalization

   In subtree generalization scheme [18] , At any node other than leaf node, either all its child values are generalized or none is generalized. For example from figure if all dancer is generalized to artist then writer have to be generalized to artist but doctor and engineer may be generalized can retain its specific value at leaf level.It is used in Global recoding algorithms.

2. Sibling Generalization

   In this generalization scheme [19],that is same as subtree generalization but in this some sibling can remain ungeneralized . For eg. if dancer is generalized to artist then writer may remain ungeneralized . It gives the lesser distortion compared to subtree and full domain and used in global recoding algorithms.

3. Cell Generalization

   All the generalization [20] schemes that are discussed earlier used, are called global recoding. They give more distortion in this scheme is a value is generalized in one record then for that specific value must be generalized in all other records also.

   But In cell generalization, it is known as local recoding there is not restriction means if a value is generalized in one record the same value for same attribute in other record may be ungeneralized. For example in a record dancer is generalized to artist dancer in other records may remain ungeneralized. The problem of this flexibility is that data utility is affected by this because while applying data mining technique in this dancer assign to class 1 and assign to class 2 so both are two different classes. While Global recoding generalizing scheme do not have this data utility problem.

1.6.2 Suppression

Suppression is similar to generalization but in this values of quasi-identifier is completely hidden for eg from sex male female to Any or not released or from specific profession to value is suppressed to not released at all. Different Supression types are defined as

1. Record Level :When the complete entry of a record from the table is eliminated or suppressed.

2. Value Level : When all instance or records of a particular value in the table is suppressed.

3. Cell Level : When some of records for a given value are suppressed in a table.

1.7 Motivation

- Individual's data collected by organizations, governments etc is increasing at day by day. In recent years, plenty of incidents have come with the cases when just removing explicit_identifier is not good enough to protect individual's privacy. Also, detailed personal specific data is often needed for research and analysis. In this scenario,data anonymization is the fundamental base for balancing an individuals privacy and providing processed data for decision making. To overcome this situation, k-anonymity is the popular technique. The main aim is to secure a given dataset against linking attack by applying anonymization operations like generalization and suppression on the quasi_identifiers. the linking attack attempts to link anonymous data to additionally publicly available data , which may cause disclosure of one's identity. [21] A given dataset is said to be k-anonymous when each of it's data item can not be distinguished from at least *k-1* other data records. In this scenario, a tabular is assumed. some other methods have been suggested (for example, differential privacy), yet k-anonymity is still preferred the first option in many fields, e.g., medicine.

1.8 Objective

Given a raw dataset D, the purpose is to transform the given dataset to an another dataset D' using anonymization operations like generalization and suppression so that the anonymized dataset D' satisfies the given privacy requirements and information loss is minimum.

1.9 Thesis Organization

Ch 1. Introduction

In this chapter we have explored briefly about data publishing and what is privacy preserving,why there is need of privacy preserving techniques while publishing data. How anonymization can be used to preserve privacy .To maintain privacy a model K anonymity is explained in it and its basic details and attack on this model.

Ch 2. Related Work

In this chapter we have discussed ,metric that are used to calculate the quality of anonymized data , the previous algoriths that have been used for k-anonymization.

Ch 3. Purposed Work

In this chapter we explained that to achieve k-anonymity, the best way is to search the lattice in the bottom-up manner using breadth first search to obtail the local optimal node.

Ch 4. Experiment Results

In the chapter we have plotted the graph ,for different values of k taken executiontime vs quasi-identifer and distortion vs quasi-identifer,.We can compare and analysis the results of our approach with previous algorithms.

Ch 5.Conclusion

In this chapter, we have explained that after comparing the results and analysis we can conclude that our purposed algorithm gives takes less time than other efficient algorithms while other metric also gives better results in maximum cases.

Chapter 2

literature Review

2.1 Metrics used to Measure the Quality of Generalized Data

Privacy preserving data publishing have two objectives, privacy of individual entity for each record must be preserved and published data must be information which is useful for data mining. So the quality of anonymized data can be measured by data metric which are classified into three categories.

2.1.1 General Purpose Metrics

When data publisher do not know what data recipient want to know or analysis from the published data so data publisher can not focus on any particular data utility [10].In this case data published is open to all like internet so that data recipient based on their different interest and they do data mining according to their requirement, in this is very obvious that same metric is not good or accurate for different recipients. In this case for better utility of anonymized data,data publisher choose metric which are more suitable for mostly all data recipients such as ILoss, distortion, discernibility.

1. ILoss

To calculate the data loss while anonymizing the data proposed a data metric known as ILoss.

$$ILoss = \frac{|Vg|-1}{|DA|}$$

Where $|Vg|$ is total number of children of node .

$|Vg|$ is total count of leaf_ nodes for that attribute having vg as a node. If ILoss = 0, means value remains ungeneralized, same as in original table.It calculates the fraction of leaf nodes that are generalized.

Example:Let a value is generalized from Lawyer to professional.

So its $ILoss = \frac{2-1}{1} = 0.25$ After generalization *ILoss* for any record can calculated as

$$ILoss(\mathrm{r}) = \sum(W_i \times ILoss(V_g))$$

W˙i is predefined weight penalty assigned to each quasi-identifer The total for complete generalized table is

$$ILoss(\mathrm{T}) = \sum_{r \epsilon T} \mathrm{ILoss(r)}$$

2 Discernibility

After anonymizing dataset ,each equivalence class has its size that is number of records in it. the class size contributes to the anonymization based on cost, it can be calculated for complete generalized dataset by this formula, Discernibility Metric

$$\mathrm{DM} = |E_i|^2$$

where E˙i is the size of equivalence class .

minimize Discernability cost leads to less distortion with is desirable requirement for better anonymiztion.

2.1.2 Special Purpose Metrics

If data publisher know for which purpose the published data will be data mined or in which information or pattern data recipient is interested ,so that they can preserve their related information and publish the data according to their requirements .For example if the purpose of data recipient is to model the classification based on a particular attribute in this case generalization must not be done for values whose identification is necessary to assign a class,which is used for their classification.

Classification Metric (CM)

Iyengar purposed a metric to measure the classification error means a record is assigned to a class by assuming that in it a particular class is not majority but in reality that class is not the majority class so, record is assigned to wrong class. There must be some penalty for it or there is a penalty if record is suppressed completely and not assigned to the any class. CM can be calculated by sum of all the penalties of each record, it is normalized by considering total number to records.

$$\text{CM} = \frac{\sum_{all\ rows} penalty(row\ r)}{N}$$

A row r is given penalty if the row is suppressed and/or if its class label class(r) is not the majority class label majority (G)of its group G.

Penality can be calculated as if a record is suppressed or it is assigned togroup assume class(r) is major class but actual that class is not the major class.

2.1.3 Trade-off Metrics

Specializing from a general value to a specific value loss some level of privacy but gain some information regarding that attribute which is specialized. Special metric while anonymizing at final information it may gain sufficient information but might lose so privacy that it is very difficult to do further anonymization. So Trade -off Metrics solve this problem, both information gain and privacy loss are calculated at every iteration of anonymization,so that optimal trade -off can be found for both necessary requirements.

In this trade-off metric [], for every specialization all records of this group are assigned to its child level group so it gain some information(IG)and as it divides the group size into smaller group there is privacy loss(PL).Objective of this metric is to find a specialization whose information gain is maximum for each privacy loss

$$\text{IGPL(s)} = \frac{IG(s)}{PL(s)+1}$$

Where IG(s) = Information gain can be decrement of class entropy or decrement of distortion by specialization.

PL(s) = $avgA(QID_j) - A_s(QID_j)$

privacyloss PL(s) = the average decrement of anonymity over all QID_j that contain the attribute of s.

A(QIDj) = the anonymity before specializing of attribute j.

2.2 Global Recording Algorithms

Datafly Algorithm

The Datafly algorithm [Sweeney (1997)] goes with the assumptions that the best solutions are the ones that are attained after generalizing the variables with the most distinct values (unique items). The search space is the whole lattice. However, this approach only goes through a few nodes in the lattice to find its solution. This approach is very efficient from a time perspective. Datafly uses a greedy algorithm to search the domain generalization hierarchy. At every step, it chooses the locally optimal move. One drawback with Datafly's approach is that it may become trapped in a local optimum [Cormen et al:(2001)].

Samarati Algorithm

Samarati algorithm assumes that the best solutions in the lattice are the ones that result in a table having minimal generalizations [Samarati (2001)] [9]. So, the solutions are available in the height that is minimal in a lattice. The algorithm is based on the axiom that if a node at level h, in domain generalization hierarchy satisfies k-anonymity, then all the levels of height higher than h also satisfy k-anonymity. In order to search the lattice and identify the the lowest level with the generalizations that satisfy k-anonymity with minimal suppression, Samarati used binary search. The algorithm goes through the lattice with a binary search, always cutting the search space in half. It goes down the level if a solution is found at that level, otherwise it goes up the lattice. Eventually, the algorithm finds the solution with the lowest height with the least generalizations. This level ensures less information loss but time consumed is higher than Datafly.

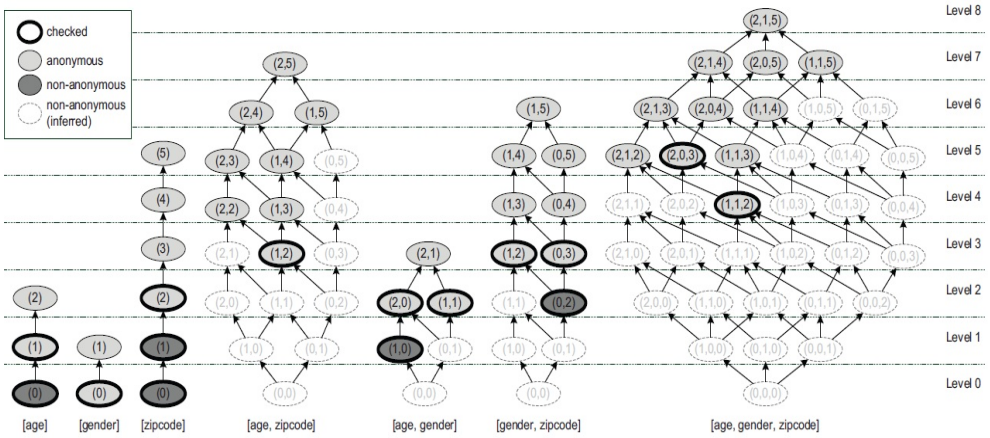

Figure 2.1: incongito algorithm example

Incognito Algorithm

Incognito [Lefevre et al: (2005)] implements a dynamic programming approach which satisfy subset property which states that a relation T can not be k-anonymous if it's subset of quasi-identifiers does not satisfy k-anonymity. The approach constructs generalization lattice of each subset of QIs and checks by performing a breadth-first bottom-up search [17]. The number of generalization lattice constructed in case of Incognito for QIs of order r is 2^r. Thus Incognito algorithm is of order (2^r) because at least one lattice is checked for k-anonymity in every generalization lattice.

Optimal Lattice Algorithm (OLA)

El Emam et al: suggested an algorithm called Optimal Lattice Anonymization and presented that it outperforms Incognito [Emam et al: (2009)]. It use predictive-tagging to reduce the search space of the lattice. However, if global optimal k-anonymous lattice lie on or above the middle level of full domain generalized hierarchy, then the algorithm check all the middle level lattices for k-anonymity. This algorithms checks only the middle level of full domain generalized hierarchy is exponential in number of QIs.

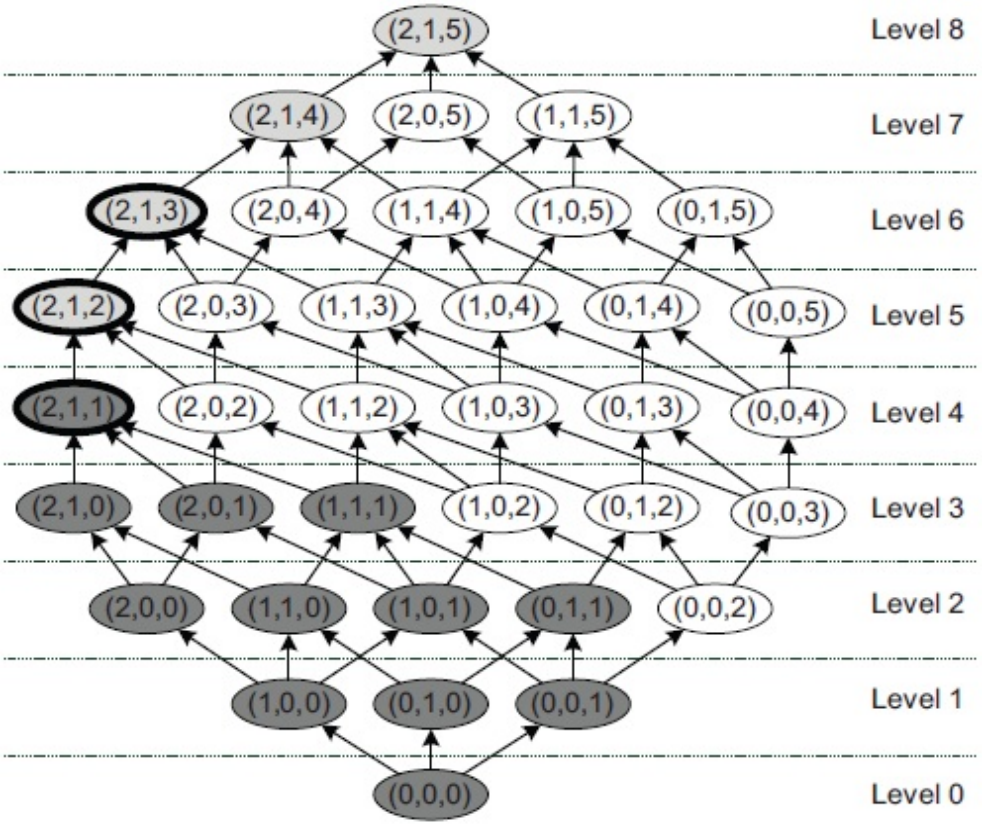

Figure 2.2: OLA example

Chapter 3

Proposed Work

3.1 Introduction

K-anonymization is a primary approach for the de-identification of datasets containing person specific information. In our work, we have described a framework to implement most of the k-anonymity algorithms and also proposed a novel scheme that produces better results with real-world datasets. The implementation framework holds complete data in main memory with dictionary compression to this data. The maximum count of QIs for the datasets considered by them is only nine. If the count of QIs is very high, then it would be difficult to put all the data items in the main memory.

3.2 Basic Framework

Our work is based on a general framework for the efficient application of k-anonymity based algorithms. In [25], we suggested an optimal and efficient application of the optimal lattice anonymization(OLA) algorithm. Furthermore, We evolute the framework in current section and outline the fundamental objective behind it:

1. The main task is to check the k-anonymous status of each state and this task should be efficient.

[17]The preliminary work of this scheme is a well planned memory layout, which allows the optimal application of various generalization schemes to a given

dataset. Additionally, the anonymization operations are problem specific. It offers some further optimization. The general implementation, involving optimization, may be applied to all global recording based anonymization schemes.

3.3 Basic Implementation

The layout contains complete data in main memory and apply dictionary compression on whole dataset. Generalization hierarchy is illustrated in a table. Generalization hierarchy for the attribute age is given in table 3.1.

Table 3.1: Tabular generalization hierarchy [1]

level 0	level 1	level 2
1	<50	*
2	<50	*
.	<50	*
.	>=50	*
99	>=50	*
100	>=50	*

A dictionary $dic_0, ..., dic_{n1}$ for each quasi identifier is required for mapping the column values with integer values [7]. Due to encoding of the given dataset values at lower level before advancing to generalization hierarchies, it is confirmed that the original dataset values for a column with m different values is given the count values 0 to m 1.To store the transformed form of the given raw dataset, a buffer data structure is used. Based on given memory framework [4], modifying an attribute value taken from the given dataset in cell (row, col) to a value described on current level of it's generalization hierarchy and collecting it in chosen buffer is described by the following assignment:

$$Buffer[row, col] \leftarrow heir_{col}data[[row, col], level]$$

While checking a particular state, one by one, the algorithm searches all rows in given dataset and for each cell it apply the above assignment .Then, the modified tuple is moved to the operator that makes equivalence classes after adding the tuple to given hash table.After that ,it is ensured whether all equivalence classes

have size greater or equal to k. Additionally a suppression value is specified which defines the upper limit on the count of rows that may be suppressed so that the dataset still remains k-anonymous. this suppression value reduces data loss. With the help of it , the classes whose size is less than k are deleted from the given dataset till the count of suppressed tuples does not exceed the defined threshold value.

3.4 Proposed approach

The proposed approach searches the generalization lattice in a bottom_up BFS manner and creates paths constantly. The scheme is focused on following asumptions:

1. Vertical traversal of the generalization lattice in binary manner utilizes predictive tagging in most efficient way.

2. The time taken in traversing the generalization lattice in vertical manner is volatile.

3. The optimal performance is achieved when algorithm utilizes the previous optimizations to check current node.

3.4.1 Main Algorithm

As given the first algorithm, it checks each node at all levels from level o to top level. It enumerates each node at each level and apply Generate_path(node) when an untagged node is found. Algorithm 2 implements a greedy approach based on depth_first_search. The searching aborts when either the algorithm reaches the top node in the lattice hierarchy or the present node does not have an untagged successor.

Algorithm 1: Main algorithm :

Input: generalized lattice

1. Create an empty min_heap
2. At each level, starting from level 0 to maxlevel ($lattice\_height - 1$)
   - 2.1: Check each node whether it is tagged or not.
   - 2.2: If not, **Generate_path** of untagged nodes starting from this node.
   - 2.3: **Check_path** for nodes that satisfies k-anonymous property.
3. until the heap is not empty:
   - 3.1: Extract_min from heap and consider it as current node.
   - 3.2: Check for each successor of this current node whether tagged or not.
   - 3.3: if not, repeat step 2.2 and 2.3 .

As the path is generated, defined function *check_path*(heap,path) is called to check the k-anonymity. As can be seen in Algorithm 3, it implements a binary search. Firstly, the node at location $\frac{1}{2}\,(path.length - 1)$ is checked .Each time,a k-anonymity check is performed, predictive tagging is applied in the lattice. On the basis of the output of the check, the algorithm moves to either lower or upper part of the generalized lattice. Each time a node is checked for k-anonymity and if resulted as non-anonymous then it is added to a min heap otherwise if found anonymous then its reference is stored to find global optimal.At each traversal, predictive tagging is taken place to reduce the search space. At the end of the search process, the optimal_node always keeps a reference of the optimal_node on each path. Finally, globally optimal_node is decided by doing a comparative study of all local optimal_nodes.

3.4.2 Algorithm 2: Generate_path(node)

Input: current_node

1. Consider the path as an empty list.
2. For each node S belonging to successor of the current_node, check whether S is tagged or not. If not

   Assign S as current_node.

   return.
3. repeat step 2 until the top node is reached or Current_node does not have an untagged successor.

After the check_path() operation the current status of the heap is taken for the further path generation starting of the successors of the found non-anonymous nodes in previous path. snapshot and roll-up optimization is the main reason behind generating these paths. The order of nodes returned by the heap is determined according to the node level in the generalized lattice. The minimal optimal node in the heap is taken in consideration in order to increase the length of the generated path that increases the chance of applying predictive tagging to the nodes in various paths in the lattice.

3.4.3 Algorithm 3:Check_path(path, heap)

Input: Heap, Generated_path

1. min $\leftarrow 0$
2. max $\leftarrow path.length - 1$
3. $optimum\_node \leftarrow NULL$
4. While min $\leq$ max do

   $mid\_node \leftarrow \frac{1}{2} \lfloor (min + max) \rfloor$

   current_node$\leftarrow$path.get(mid)
5. If check_and_tagged(current_node)

   $optimum\_node \leftarrow current\_node$

   max $\leftarrow mid\_node - 1$

6. else

   heap.add(current_node)

   min$\leftarrow$ $mid\_node + 1$

7. Store optimum_node

When the heap becomes empty, the main algorithm halts with the termination of the outer loop of the main algorithm.

Chapter 4

Implementation and Results

4.1 Implementation Setup and used Dataset

Implementation is done on System having configuration intel (R) $core(TM)$ i5-3210m CPU @ 2.50ghz, 4GB RAM. Our Implementation is done PYTHON IDLE 2.7.6 .Complete Adult Data Set which contains 32,561 records is taken for analysis results.The attributes for quasi identifier are Age which is numeric, Work class which is categorical, Education which is categorical, Marital status is categorical, race which is categorical, gender is categorical, Occupation and salary are sensitive attributes. We have taken Discernibility Metric and Exceution Time as parameters to evaluate and analyse the result for k values taken as 2, 5, 10 over the proposed algorithm and other previous algorithms like samarati,incognito, OLA(Optimal lattice Anonymization).

S.No	Attributes	Generalizations	Distinct Value	Height
1	Work Class	Taxonomy Tree	7	3
2	Education	Taxonomy Tree	16	4
3	Marital Status	Taxonomy Tree	7	3
4	Race	Taxonomy Tree	5	2
5	Sex	Suppression	2	1
6	Occupation	Taxonomy Tree	14	2
7	Salary	Suppression	2	1

Table 4.1: Description of Adult Dataset

4.1.1 Discernibility Metric

We used Discernibility Metric to measure the quality of anonymized data , the lesser is discerniblity cost ,better is the quality is anonymized Data . By refering figures Figure-4.2 ,Figure-4.4 we can conclude that For smaller K value k=2,5 and , for all number quasi-identifers taken our approach give better anonymized data than incognito, Samarati and OlA algorithm and if K is large, K= 10 and number of quasi identifier taken not large our approach gives lesser discernibility otherwise gives similar result.

4.1.2 Execution Time

We considered Execution time also to evaluate and compare our approach with Incognito and Samarati and OLA.By refering figures Figure-4.1 , Figure-4.3, Figure-4.5, we can conclude that for all k values 2, 5, 10 and our approach take lesser execution time than Incognito, Samarati,OLA algorithm. For all k values taken and for all number of quasi identifier taken so we can conclude our approach is faster compared to others.

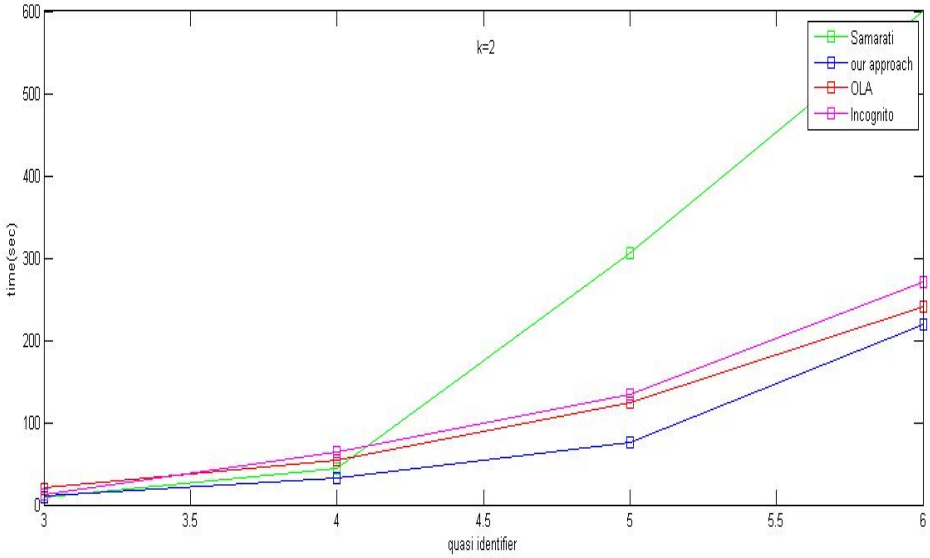

Figure 4.1: Execution time(sec) VS Quasi-identifier

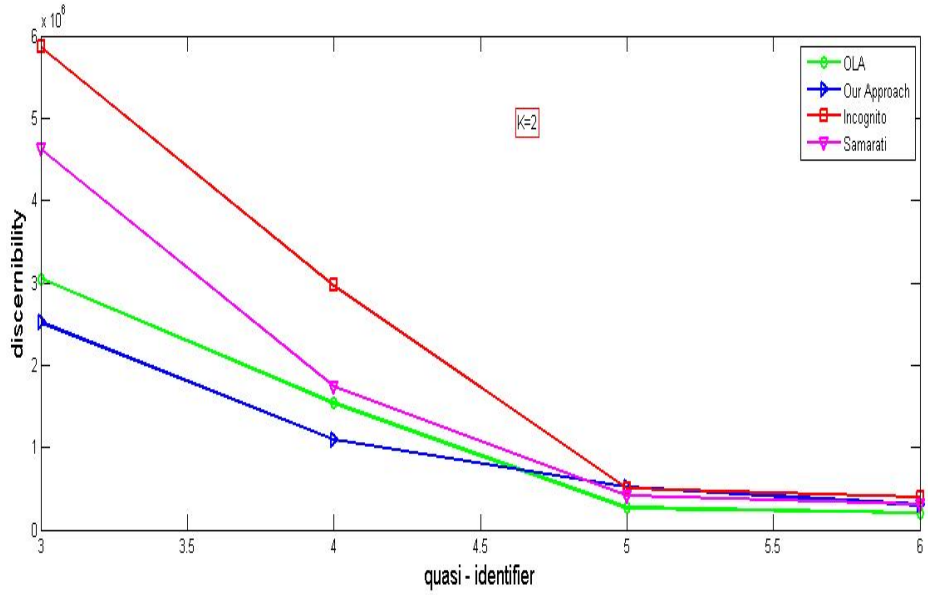

Figure 4.2: Discernibility vs Quasi-Identifier

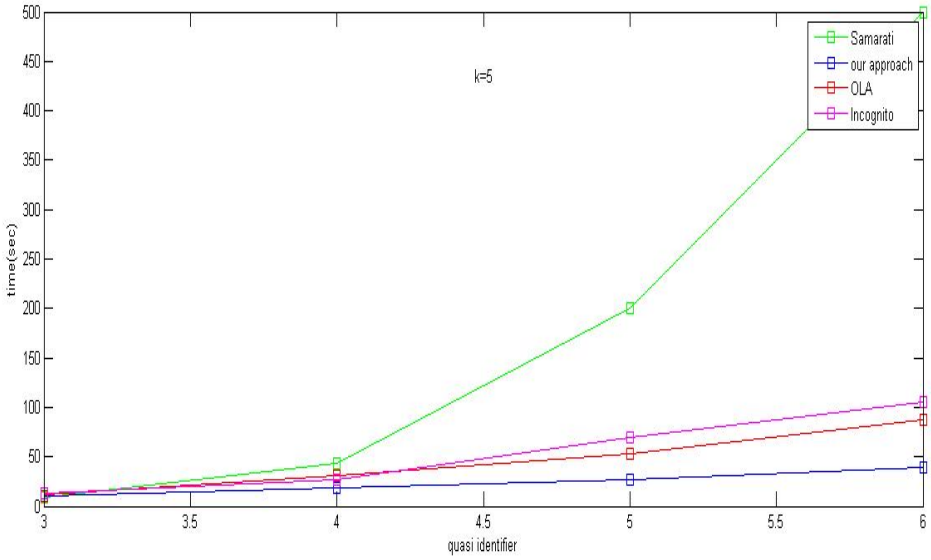

Figure 4.3: Execution time(sec) VS Quasi-identifier

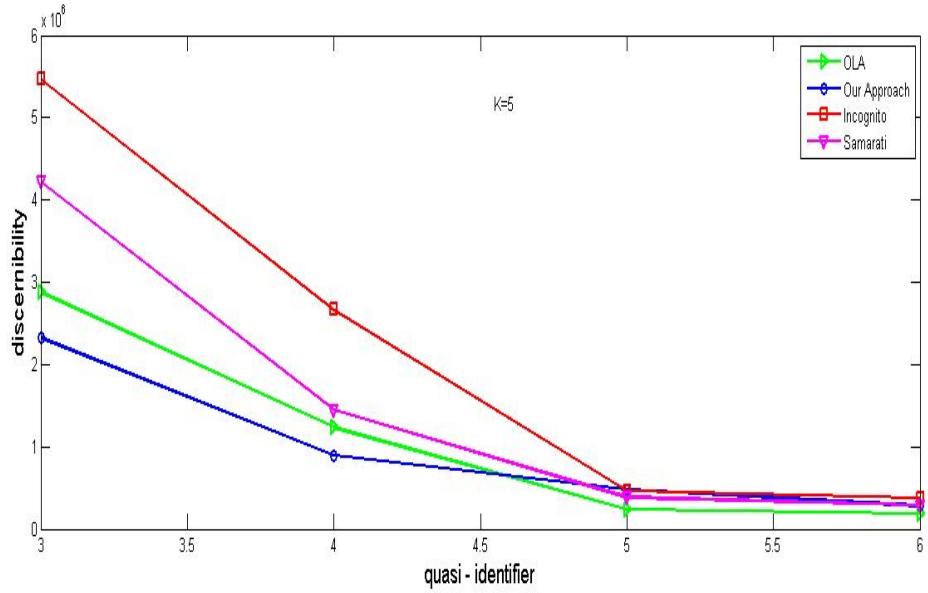

Figure 4.4: Discernibility vs Quasi-identifier

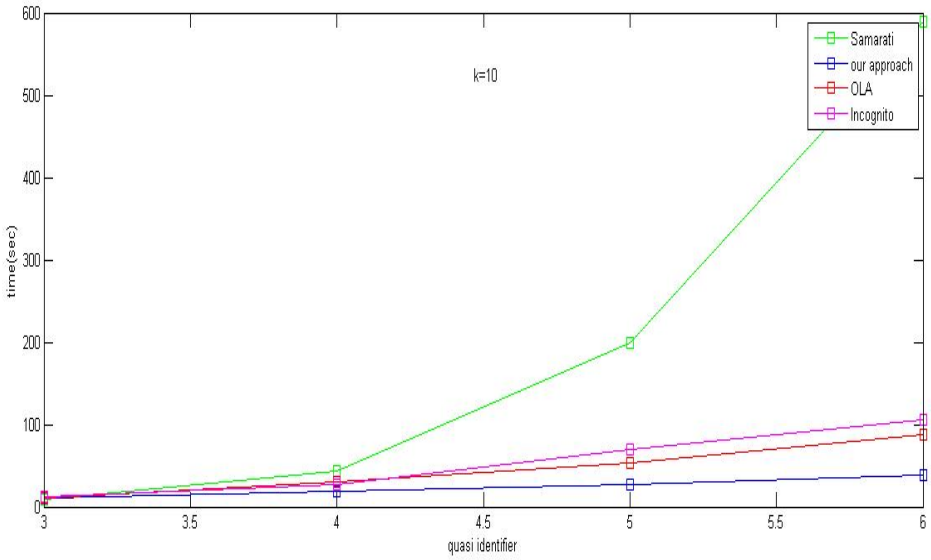

Figure 4.5: Execution time(sec) VS Quasi-identifier

Chapter 5

Conclusion and Future Work

In our work, we have described a framework to implement most of the k-anonymity algorithms and also proposed a novel scheme that produces better results with real-world datasets. Further we explained that the framework is applicable for the implementation of k-anonymity schemes like Incognito, Samarati, Datafly and optimal lattice Anonymization(OLA).Further we shown that optimal lattice anonymization performs better than incognito.We further proposed a generic k-anonymization scheme which gives better result than Incognito, Samarati, OLA, Datafly. As it traverse the lattice vertically, it utilizes the predictive tagging in best way to make extensive use of the proposed layout.

In future ,The algorithms discussed in this thesis can be further improved by reducing the size of the solution space and applying improved searching algorithms. In future there may be need of disk based application of k-anonymity algorithm because of limited main memory space.

Bibliography

[1] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," in *PODS*, vol. 98, p. 188, 1998.

[2] B. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys (CSUR)*, vol. 42, no. 4, p. 14, 2010.

[3] Y. Yuan, J. Yang, J. Zhang, S. Lan, and J. Zhang, "Evolution of privacy-preserving data publishing," in *Anti-Counterfeiting, Security and Identification (ASID), 2011 IEEE International Conference on*, pp. 34–37, IEEE, 2011.

[4] J. Gehrke, "Models and methods for privacy-preserving data analysis and publishing," in *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pp. 105–105, IEEE, 2006.

[5] A. Gionis and T. Tassa, "k-anonymization with minimal loss of information," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 2, pp. 206–219, 2009.

[6] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 571–588, 2002.

[7] J. Goldberger and T. Tassa, "Efficient anonymizations with enhanced utility," in *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pp. 106–113, IEEE, 2009.

[8] M. Hua and J. Pei, "A survey of utility-based privacy-preserving data transformation methods," in *Privacy-Preserving Data Mining*, pp. 207–237, Springer, 2008.

[9] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," tech. rep., Technical report, SRI International, 1998.

[10] B. C. Fung, K. Wang, A. W.-C. Fu, and S. Y. Philip, *Introduction to privacy-preserving data publishing: concepts and techniques*. CRC Press, 2010.

[11] R. C.-W. Wong, A. W.-C. Fu, K. Wang, and J. Pei, "Anonymization-based attacks in privacy-preserving data publishing," *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 2, p. 8, 2009.

[12] Z. FeiFei, D. LiFeng, W. Kun, and L. Yang, "Study on privacy protection algorithm based on k-anonymity," *Physics Procedia*, vol. 33, pp. 483–490, 2012.

[13] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[14] V. S. Iyengar, "Transforming data to satisfy privacy constraints," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 279–288, ACM, 2002.

[15] G. V. Kanth and B. S. Kumar, "A study of novel anonymization techniques for secure data publishing," *International Journal of Engineering*, vol. 2, no. 5, 2013.

[16] K. LeFevre, 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*, pp. 49–60, ACM, 2005.

[17] A. Meyerson and R. Williams, "On the complexity of optimal k-anonymity," in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 223–228, ACM, 2004.

[18] S. K. Adusumalli and V. V. Kumari, "Attribute based anonymity for preserving privacy," in *Advances in Computing and Communications*, pp. 572–579, Springer, 2011.

[19] B. Berčič and C. George, "Identifying personal data using relational database design principles," *International Journal of Law and Information Technology*, vol. 17, no. 3, pp. 233–251, 2009.

[20] K. El Emam, F. K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J.-P. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, *et al.*, "A globally optimal k-anonymity method for the de-identification of health data," *Journal of the American Medical Informatics Association*, vol. 16, no. 5, pp. 670–682, 2009.

[21] P. Shi, L. Xiong, and B. Fung, "Anonymizing data with quasi-sensitive attribute values," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1389–1392, ACM, 2010.