Volume 4, Issue 3, March 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

An Efficient Sliced Data Algorithm Design for Personalized Data Protection to Prevent Generalized Losses and Membership Divulgence

Amar Paul Singh

Dept. of Computer Application

G.C.Ghumarwin Himachal Pradesh India

Abstract: - In this information age, data and knowledge extracted by data mining techniques represent a key asset driving research, innovation, and policy-making activities. Many agencies and organizations have recognized the need of accelerating such trends and are therefore willing to release the data they collected to other parties, for purposes such as research and the formulation of public policies. However the data publication processes are today still very difficult. Data often contains personally identifiable information and therefore releasing such data may result in privacy breaches, this is the case for the examples of micro-data, e.g., census data and medical data. This thesis studies how we can publish and share micro data in a privacy-preserving manner. This present an extensive study of this problem along three dimensions: Designing a simple, intuitive, and robust privacy model, Designing an effective anonymization technique that works on sparse and high-dimensional data and developing a methodology for evaluating privacy and utility tradeoffs.

Keywords- Data Anonymization, Micro Data, Privacy Preserving Data Publishing, Slicing.

I. INTRODUCTION

Data Anonymization is a technology that convert clear text into a non-human readable form.Data anonymization technique for privacy-preserving data publishing has received a lot of attention in recent years. Detailed data (also called as micro-data) contains information about a person, a household or an organization. Most popular anonymization techniques are Generalization and Bucketization. [1]There are number of attributes in each record which can be categorized as 1) Identifiers such as Name or Social Security Number are the attributes that can be uniquely identify the individuals. 2) some attributes may be Sensitive Attributes(SAs) such as disease and salary and 3) some may be Quasi-Identifiers (QI) such as zipcode, age, and sex whose values, when taken together, can potentially identify an individual.
Data anonymization enables the transfer of information across a boundary, such as between two departments within an agency or between two agencies, while reducing the risk of unintended disclosure, and in certain environments in a manner that enables evaluation and analytics post-anonymization

Various Anonymization Techniques:

Two widely studied data anonymization technique are generalization and bucketization. The main difference between the two anonymization techniques lies in that bucketization does not generalize the QI attributes.

- **Generalization**

Generalization is one of the commonly anonymized approach, which replaces quasi-identifier values with values that are less-specific but semantically consistent. Then, all quasi-identifier values in a group would be generalized to the entire group extent in the QID space. [12] If at least two transactions in a group have distinct values in a certain column (i.e. one contains an item and the other does not), then all information about that item in the current group is lost. The QID used in this process includes all possible items in the log. Due to the high-dimensionality of the quasi-identifier, with the number of possible items in the order of thousands, it is likely that any generalization method would incur extremely high information loss, rendering the data useless [8]. In order for generalization to be effective, records in the same bucket must be close to each other so that generalizing the records would not lose too much information. However, in high-dimensional data, most data points have similar distances with each other. To perform data analysis or data mining tasks on the generalized table, the data analyst has to make the uniform distribution assumption that every value in a generalized interval/set is equally possible, as no other distribution assumption can be justified. This significantly reduces the data utility of the generalized data. And also because each attribute is generalized separately, correlations between different attributes are lost. In order to study attribute correlations on the generalized table, the data analyst has to assume that every possible combination of attribute values is equally possible. This is an inherent problem of generalization that prevents effective analysis of attribute correlations.

- **Bucketization**

Bucketization The first, which we term bucketization, is to partition the tuples in T into buckets, and then to separate the sensitive attribute from the non-sensitive ones by randomly permuting the sensitive attribute values within each bucket.

The sanitized data then consists of the buckets with permuted sensitive values. In this paper,[13,] we use bucketization as the method of constructing the published data from the original table T, although all our results hold for full-domain generalization as well. We now specify our notion of bucketization more formally. Partitition the tuples into buckets (i.e., horizontally partition the table T according to some scheme), and within each bucket, we apply an independent random permutation to the column containing S-values. The resulting set of buckets, denoted by B, is then published. For example, if the underlying table T, then the publisher might publish bucketization B .Of course, for added privacy, the publisher can completely mask the identifying attribute (Name) and may partially mask some of the other non-sensitive attributes (Age, Sex, Zip). While bucketization [1], [13] has better data utility than generalization, it has several limitations. First, bucketization does not prevent membership disclosure. Because bucketization publishes the QI values in their original forms, an adversary can find out whether an individual has a record in the published data or not. As shown in, 87 percent of the individuals in the United States can be uniquely identified using only three attributes (Birthdate, Sex, and Zip code). A micro-data (e.g., census data) usually contains many other attributes besides those three attributes. This means that the membership information of most individuals can be inferred from the bucketized table. Second, bucketization requires a clear separation between QIs and SAs. However, in many data sets, it is unclear which attributes are QIs and which are SAs. Third, by separating the sensitive attribute from the QI attributes, bucketization breaks the attribute correlations between the QIs and the SAs. Bucketization first partitions tuples in the table into buckets and then separates the quasi identifiers with the sensitive attribute by randomly permuting the sensitive attribute values in each bucket. The anonymized data consist of a set of buckets with permuted sensitive attribute values. In particular, bucketization has been used for anonym zing high-dimensional data. However, their approach assumes a clear separation between QIs and SAs. In addition, because the exact values of all QI's are released, membership information is disclosed.

II. BACKGROUND

Privacy-Preserving Data Publishing

Figure 1.2: A Simple Model of PPDP [13].

Two main Privacy preserving paradigms have been established: k-anonymity [7], which prevents identification of individual records in the data, and l-diversity [1], which prevents the association of an individual record with a sensitive attribute value.

- **K-anonymity**

The database is said to be K-anonymous where attributes are suppressed or generalized until each row is identical with at least k-1 other rows. K-Anonymity thus prevents definite database linkages. K-Anonymity guarantees that the data released is accurate. K-anonymity proposal focuses on two techniques in particular, generalization and suppression [2]. To protect respondents' identity when releasing micro-data, data holders often remove or encrypt explicit identifiers, such as names and social security numbers. De-identifying data, however, provide no guarantee of anonymity. Released information often contains other data, such as birth date, sex, and ZIP code that can be linked to publicly available information to re-identify respondents and to infer information that was not intended for release. One of the emerging concept in micro-data protection is k-anonymity, which has been recently proposed as a property that captures the protection of a micro-data table with respect to possible re-identification of the respondents to which the data refer. K-anonymity demands that every tuple in the micro-data table released be indistinguishably related to no fewer than k respondents. One of the interesting aspect of k-anonymity is its association with protection techniques that preserve the truthfulness of the data. The first approach toward privacy protection in data mining was to perturb the input (the data) before it is mined. The drawback of the perturbation approach is that it lacks a formal framework for proving how much privacy is guaranteed. At the same time, a second branch of privacy preserving data mining was developed, using cryptographic techniques. Thus, it falls short of providing a complete answer to the problem of privacy preserving data mining. One definition of privacy which has come a long way in the public arena and is accepted today by both legislators and corporations is that of k-anonymity [3]. The guarantee given by k-anonymity is that no information can be linked to groups of less than k individuals. Generalization for k-anonymity losses considerable amount of information, especially for high-dimensional data.

Limitations of k-anonymity are: (1) it does not hide whether a given individual is in the database, (2) it reveals individuals' sensitive attributes , (3) it does not protect against attacks based on background knowledge , (4) mere knowledge of the k-anonymization algorithm can violate privacy, (5) it cannot be applied to high-dimensional data without complete loss of utility , and (6) special methods are required if a dataset is anonymized and published more than once.

l- Diversity

The next concept is "l-diversity". Say you have a group of k different records that all share a particular quasi-identifier. That's good, in that an attacker cannot identify the individual based on the quasi-identifier. But what if the value they're interested in, (e.g. the individual's medical diagnosis) is the same for every value in the group. The distribution of target values within a group is referred to as "l-diversity". [8] Currently, there exist two broad categories of l-diversity techniques: generalization and permutation-based. An existing generalization method would partition the data into disjoint groups of transactions, such that each group contains sufficient records with l-distinct, well represented sensitive items.

III. PROBLEM DEFINITION

- **Privacy Models**

A number of privacy models have been proposed in the literature, including k-anonymity and ℓ-diversity.

K-Anonymity: - Samarati and Sweeney [1, 4, 10] introduced *k-anonymity* as the property that each record is indistinguishable with at least *k-1* other records with respect to the quasi-identifier. In other words, k-anonymity requires that each QI group contains at least k records.

ℓ-**Diversity: -** While k-anonymity protects against identity disclosure, it does not provide sufficient protection against attribute disclosure.

- **Anonymization Methods.**

Generalization replaces a value with a "less-specific but semantically consistent" value. Tuple suppression removes an entire record from the table. Unlike traditional privacy protection techniques such as data swapping and adding noise, information in a k-anonymized table through generalization and suppression remains truthful.

Bucketization Another anonymization method is bucketization (also known as anatomy or permutation-based anonymization) [20, 21]. The bucketization method first partitions tuples in the table into buckets and then separates the quasi-identifiers with the sensitive attribute by randomly permuting the sensitive attribute values in each bucket. The anonymized data consists of a set of buckets with permuted sensitive attribute values.

- **Data Utility Measures**

it can trivially anonymize the data by removing all quasi-identifiers. This provides maximum privacy and the data becomes useless. The only reason to publish and share data is to allow research and analysis on the data. It is important to measure the utility of the anonymized data.

- **Limitations of Current Privacy Principles.**

Limitation of p-sensitive k-anonymity: The purpose of *p*-sensitive *k*-anonymity is to protect against attribute disclosure by requiring that there be at least *p* different values for each sensitive attribute within the records sharing a combination of quasi-identifier. This approach has the limitation of implicitly assuming that each sensitive attribute takes values uniformly over its domain, that is, that the frequencies of the various values of a confidential attribute are similar. When this is not the case, achieving the required level of privacy may cause a huge data utility loss.

Limitation of l-diversity: The *l*-diversity model protects against sensitive attribute disclosure by considering the distribution of the attributes. The approach requires *l* \well-represented" 1 values in each combination of quasi-identifiers. This may be difficult to achieve and, like *p*-sensitive *k*-anonymity, may result in a large data utility loss. Further, as previously identified, *l*-diversity is insufficient to prevent similarity attack.

- **Existing System:**

First, many existing clustering algorithms (e.g., k- means) requires the calculation of the "centroids". But there is no notion of"centroids" in our setting where each attribute forms a data point in the clustering space. Second, k-medoid method is very robust to the existence of outliers (i.e., data points that are very far away from the rest of data points). Third, the order in which the data points are examined does not affect the clusters computed from the k-medoid method.

Disadvantages:

1. Existing anonymization algorithms can be used for column generalization, e.g., Mondrian. The algorithms can be applied on the sub-table containing only attributes in one column to ensure the anonymity requirement.
2. Existing data analysis (e.g., query answering) methods can be easily used on the sliced data.
3. Existing privacy measures for membership disclosure protection include differential privacy and presence.

- **Proposed System**

This paper present a novel technique called slicing, which partitions the data both horizontally and vertically. We show that slicing preserves better data utility than generalization and can be used for membership disclosure protection. Another important advantage of slicing is that it can handle high-dimensional data. We show how slicing can be used for attribute disclosure protection and develop an efficient algorithm for computing the sliced data that obey the ℓ-diversity requirement. This paper confirm that slicing preserves better utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute.

Advantages:

1. A novel data anonymization technique called slicing to improve the current state of the art.
2. Slicing can be effectively used for preventing attribute disclosure, based on the privacy requirement of ℓ-diversity.
3. An efficient algorithm for computing the sliced table that satisfies ℓ-diversity. Our algorithm partitions attributes into columns, applies column generalization, and partitions tuples into buckets. Attributes that are highly-correlated are in the same column.
4. Slicing preserves much better data utility than generalization. In workloads involving the sensitive attribute, slicing is also more effective than bucketization. In some classification experiments, slicing shows better performance than using the original data (which may overfit the model). It also shows the limitations of bucketization in membership disclosure protection and slicing remedies these limitations.

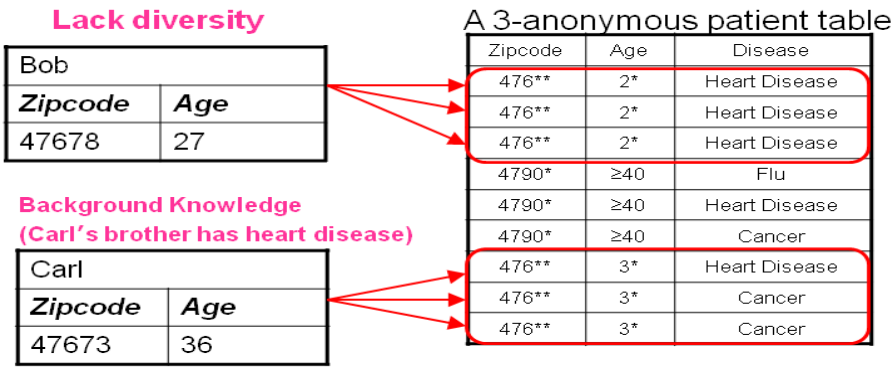

Figure 1.2 System Architecture [12].

IV. SLICED DATA ALGORITHM

Slicing

To improve the current state of the art in this paper, we introduce a novel data anonymization technique called slicing. [1] Slicing partitions the data set both vertically and horizontally. Vertical partitioning is done by grouping attributes into columns based on the correlations among the attributes. Each column contains a subset of attributes that are highly correlated. Horizontal partitioning is done by grouping tuples into buckets. Finally, within each bucket, values in each column are randomly permutated (or sorted) to break the linking between different columns. The basic idea of slicing is to break the association cross columns, but to preserve the association within each column. This reduces the dimensionality of the data and preserves better utility than generalization and bucketization. Slicing preserves utility because it groups highly correlated attributes together, and preserves the correlations between such attributes. Slicing protects privacy because it breaks the associations between uncorrelated attributes, which are infrequent and thus identifying. Note that when the data set contains QIs and one SA, bucketization has to break their correlation; slicing, on the other hand, can group some QI attributes with the SA, preserving attribute correlations with the sensitive attribute.

The key intuition that slicing provides privacy protection is that the slicing process ensures that for any tuple, there are generally multiple matching buckets. Slicing first partitions attributes into columns. Each column contains a subset of attributes. Slicing also partition tuples into buckets. Each bucket contains a subset of tuples. This horizontally partitions the table. Within each bucket, values in each column are randomly permutated to break the linking between different columns.

This algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning. We now describe the three phases.

- **Attribute Partitioning**

This algorithm partitions attributes so that highly correlated attributes are in the same column. This is good for both utility and privacy. In terms of data utility, grouping highly correlated attributes preserves the correlations among those attributes. In terms of privacy, the association of uncorrelated attributes presents higher identification risks than the association of highly correlated attributes because the associations of uncorrelated attribute values is much less frequent and thus more identifiable.

- **Column Generalization**

First, column generalization may be required for identity/membership disclosure protection. If a column value is unique in a column, a tuple with this unique column value can only have one matching bucket. This is not good for privacy

protection, as in the case of generalization/bucketization where each tuple can belong to only one equivalence-class/bucket.

- **Tuple Partitioning**

The algorithm maintains two data structures: 1) a queue of buckets Q and 2) a set of sliced buckets SB. Initially, Q contains only one bucket which includes all tuples and SB is empty. For each iteration, the algorithm removes a bucket from Q and splits the bucket into two buckets [5]. If the sliced table after the split satisfies l-diversity, then the algorithm puts the two buckets at the end of the queue Q Otherwise, we cannot split the bucket anymore and the algorithm puts the bucket into SB.When Q becomes empty, we have computed the sliced table. The set of sliced buckets is SB.

Slicing Architecture

Generally in privacy preservation there is a loss of security. The privacy protection is impossible due to the presence of the adversary's background knowledge in real life application. Data in its original form contains sensitive information about individuals. These data when published violate the privacy. The current practice in data publishing relies mainly on policies and guidelines as to what types of data can be published and on agreements on the use of published data. The approach alone may lead to excessive data distortion or insufficient protection. Privacy-preserving data publishing (PPDP) provides methods and tools for publishing useful information while preserving data privacy. Many algorithms like bucketization, generalization have tried to preserve privacy however they exhibit attribute disclosure. So to overcome this problem an algorithm called slicing is used.

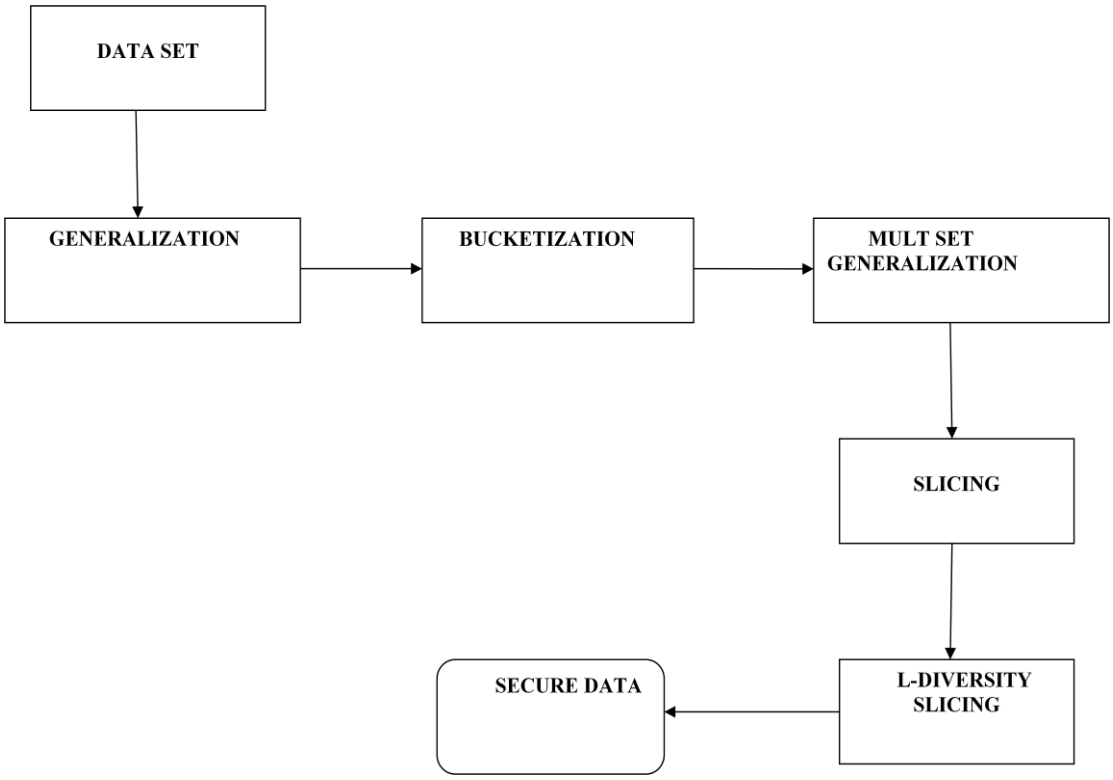

Figure 1.3 Architecture of Sliced Data Algorithm.

Algorithms used as a sliced data Algorithm

Algorithm 1

1. Q = {T}, SB = ϕ.
2. While Q is not empty
3. Remove the first bucket B from Q, Q = Q − {B}.
4. Split B into two buckets B1 and B2, as in Mondrian.
5. If diversity-check (T, Q ∪ {B1, B2} ∪ SB, ℓ)
6. Q = Q ∪ {B1, B2}.
7. Else SB = SB ∪ {B}.
8. Return SB.

Algorithm 1.1

1. For each tuple t ∈ T, L[t] = ϕ.
2. For each buckets B in T*

3. Record f (v) for each column value v in bucket B.
4. for each tuple t ∈ T
5. Calculate p(t,B) and find D(t,B).
6. L[t] = L[t] ∪ {hp (t, B), D (t, B) i}.
7. for each tuple t ∈ T
8. Calculate p(t, s) for each s based on L[t].
9. If p(t, s) ≥ 1/ℓ, return false.
10. Return true.

V. IMPLEMENTATION

Module Description

1. Original Data
2. Generalized Data
3. Bucketized Data
4. Multiset-based Generalization Data
5. One-attribute-per-Column Slicing Data
6. Sliced Data

- **Original Data**

It conducts extensive workload experiments. This Thesis results confirm that slicing preserves much better data utility than generalization. In workloads involving the sensitive attribute, slicing is also more effective than bucketization. In some classification experiments, slicing shows better performance than using the original data.

Table 1.1 Original Data For Sliced Data Algorithm.

PatientID	Name	Email	Mobile	DOB	Age	Gender	Zipcode
293	Mani	mani@gmail.com	9090909098	12/09/1999	25	Male	600024
592	Mehala	mehala@gmail.com	9098909890	12/09/1987	25	Female	600023
340	Raji	raji@gmail.com	9888877770	30/09/1988	28	Female	600023
604	gmeet	gmeetg@gmail.com	9815216606	05-01-1981	32	Male	160035
661	Cool	cool@gmail.com	9898989090	23/07/1987	32	Male	600022
69	Mahesh	mahesh@gmail.com	9000088889	27/04/1978	38	Male	600026
743	Kings	kings@gmail.com	9666677788	23/12/1978	43	Male	600023
889	Amar	amar@gmail.com	9008890088	23/06/1986	55	Male	600024
603	Karthick	karthick@gmail.com	9790989097	22/02/1984	56	Male	600054

- **Generalized Data**

Generalized Data, in order to perform data analysis or data mining tasks on the generalized table, the data analyst has to make the uniform distribution assumption that every value in a generalized interval/set is equally possible, as no other distribution assumption can be justified. This significantly reduces the data utility of the generalized data.

Table 1.2 Generalized Data For Sliced Data Algorithm.

Patient-ID	Name	Email	Disease	DOB	Age	Gender	Zipcode
293	Mani	mani@gmail.com	dyspepsia	12/09/19**	1-30	*	600***
340	Raji	raji@gmail.com	bronchitis	30/09/19**	1-30	*	600***
592	Mehala	mehala@gmail.com	flu	12/09/19**	1-30	*	600***
231	Kalai	kalai@gmail.com	gastritis	24/02/19**	31-60	*	600***
889	Amar	amar@gmail.com	flu	23/06/19**	31-60	*	600***
743	Kings	kings@gmail.com	bronchitis	23/12/19**	31-60	*	600***
694	Palani	palani@gmail.com	dyspepsia	21/12/19**	31-60	*	600***
696	Mahesh	mahesh@gmail.com	fever	27/04/19**	31-60	*	600***
661	Cool	cool@gmail.com	BP	23/07/19**	31-60	*	600***
604	gmeet	gmeetg@gmail.com	qwa	05-01-19**	31-60	*	160***

603	Karthick	karthick@gmail.com	fever	22/02/19**	31-60	*	600***
692	Sathiya	sathiya@gmail.com	gastritis	23/06/19**	61-90	*	600***
914	Balaji	balaji@gmail.com	flu	12/12/19**	61-90	*	600***

- **Bucketized Data**

This thesis work shows the effectiveness of slicing in membership disclosure protection. For this purpose, we count the number of fake tuples in the sliced data. This Work also compares the number of matching buckets for original tuples and that for fake tuples. This thesis work experiment results show that bucketization does not prevent membership disclosure as almost every tuple is uniquely identifiable in the bucketized data.

Table 1.3 Bucketized Data Table.

Patient-ID	Email	Disease	DOB	Age	Gender	Zipcode
293	mani@gmail.com	dyspepsia	12/09/1999	*	Male	600024
592	mehala@gmail.com	flu	12/09/1987	*	Female	600023
340	raji@gmail.com	bronchitis	30/09/1988	*	Female	600023
604	gmeetg@gmail.com	qwa	05/01/1981	*	Male	160035
661	cool@gmail.com	BP	23/07/1987	*	Male	600022
693	mahesh@gmail.com	fever	27/04/1978	*	Male	600026
743	kings@gmail.com	bronchitis	23/12/1978	*	Male	600023
889	amar@gmail.com	flu	23/06/1986	*	Male	600024
603	karthick@gmail.com	fever	22/02/1984	*	Male	600054
694	palani@gmail.com	dyspepsia	21/12/1965	*	Male	600025
692	sathiya@gmail.com	gastritis	23/06/1967	*	Female	600033
918	balaji@gmail.com	flu	12/12/1912	*	Male	600032

- **Multiset-based Generalization Data**

This work observe that this Multiset-based generalization is equivalent to a trivial slicing scheme where each column contains exactly one attribute, because both approaches preserve the exact values in each attribute but break the association between them within one bucket.

Table 1.4 Multiset Value For Sliced Data Algorithm.

Patient-ID	Name	Email	Disease	DOB	Age	Gender	Zipcode
293	Mani	mani@gmail.com	dy	12/09/1999	25	Male	600024
592	Mehala	mehala@gmail.com	fl	12/09/1987	25	Female	600023
340	Raji	raji@gmail.com	br	30/09/1988	28	Female	600023
604	gmeet	gmeetg@gmail.com	qw	05-01-1981	32	Male	160035
661	Cool	cool@gmail.com	BP	23/07/1987	32	Male	600022
69	Mahesh	mahesh@gmail.com	fe	27/04/1978	38	Male	600026
743	Kings	kings@gmail.com	br	23/12/1978	43	Male	600023
889	Amar	amar@gmail.com	fl	23/06/1986	55	Male	600024
603	Karthick	karthick@gmail.com	fe	22/02/1984	56	Male	600054
694	Palani	palani@gmail.com	dy	21/12/1965	56	Male	600025
692	Sathiya	sathiya@gmail.com	ga	23/06/1967	65	Female	600033
91	Balaji	balaji@gmail.com	fl	12/12/1912	68	Male	600032

- **One-attribute-per-Column Slicing Data**

This Thesis work observe that while one-attribute-per-column slicing preserves attribute distributional information, it does not preserve attribute correlation, because each attribute is in its own column. In slicing, one group correlated attributes together in one column and preserves their correlation. For example, in the sliced table shown in Table correlations between Age and Sex and correlations between Zip code and Disease are preserved. In fact, the sliced table encodes the same amount of information as the original data with regard to correlations between attributes in the same column.

Table 1.5 One Column Slicing.

Patient-ID	Name	Email	Mobile	DOB	Age	Gender	Zipcode
692	Sathiya	sathiya@gmail.com	null	23/06/1967	65	Female	600033
592	Mehala	mehala@gmail.com	9098909890	12/09/1987	25	Female	600023
889	Amar	amar@gmail.com	9008890088	23/06/1986	55	Male	600024
340	Raji	raji@gmail.com	9888877770	30/09/1988	28	Female	600023
743	Kings	kings@gmail.com	9666677788	23/12/1978	43	Male	600023
91	Balaji	balaji@gmail.com	9667788990	12/12/1912	68	Male	600032
293	Mani	mani@gmail.com	9090909098	12/09/1999	25	Male	600024
69	Mahesh	mahesh@gmail.com	9000088889	27/04/1978	38	Male	600026
603	Karthick	karthick@gmail.com	9790989097	22/02/1984	56	Male	600054
661	Cool	cool@gmail.com	9898989090	23/07/1987	32	Male	600022
694	Palani	palani@gmail.com	9777788888	21/12/1965	56	Male	600025
604	gmeet	gmeetg@gmail.com	9815216606	05-01-1981	32	Male	160035

- **Sliced Data**

Another important advantage of slicing is its ability to handle high-dimensional data. By partitioning attributes into columns, slicing reduces the dimensionality of the data. Each column of the table can be viewed as a sub-table with a lower dimensionality. Slicing is also different from the approach of publishing multiple independent sub-tables in that these sub-tables are linked by the buckets in slicing.

Table 1.6 Sliced Data Table

Patient-ID,DYS	Name,Age	Gender	Zipcode
(293,dysp)	(Mani,1-30)	(M)	(600024)
(340,bron)	(Raji,1-30)	(F)	(600023)
(592,flu)	(Mehala,1-30)	(F)	(600023)
(231,gast)	(Kalai,31-60)	(M)	(600032)
(889,flu)	(Amar,31-60)	(M)	(600024)
(743,bron)	(Kings,31-60)	(M)	(600023)
(694,dysp)	(Palani,31-60)	(M)	(600025)
(69,feve)	(Mahesh,31-60)	(M)	(600026)
(661,BP)	(Cool,31-60)	(M)	(600022)
(604,qwa)	(gmeet,31-60)	(M)	(160035)
(603,feve)	(Karthick,31-60)	(M)	(600054)
(692,gast)	(Sathiya,61-90)	(F)	(600033)
(91,flu)	(Balaji,61-90)	(M)	(600032)

VI. CONCLUSION

Slicing is a better technique for Privacy Preserving Data Publishing. It is a advancement over data Anonymization techniques. Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats, slicing where each attribute is in exactly one column. An extension is the notion of overlapping slicing, which duplicates an attribute in more than one column. Our experiments show that random grouping is not very effective. The Proposed grouping algorithm is optimized L-diversity slicing check algorithm that Provides secure data. Another direction is to design data mining tasks using the anonymized data computed by various anonymization techniques. Another important advantage of slicing is that it can handle high-dimensional data.

REFERENCES

[1] Tiancheng Li, Ninghui Li, Senior Member, IEEE, Jia Zhang, Member, IEEE, and Ian Molloy "Slicing: A New Approach for Privacy Preserving Data Publishing" Proc. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 3, MARCH 2012.

[2] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati On K-Anonymity. In Springer US, Advances in Information Security (2007).

[3] Latanya Sweeney. k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5):557–570, 2002.

[4] J. Brickell and V. Shmatikov, "The Cost of Privacy: Destruction of Data-Mining Utility in Anonymized Data Publishing," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 70-78, 2008

[5] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and „-Diversity," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), pp. 106-115, 2007.

[6] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. "l-diversity: Privacy beyond k-anonymity". In ICDE, 2006.

[7] D. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Halpern. "Worst-case background knowledge for privacy-preserving data publishing". In ICDE, 2007.

[8] G.Ghinita, Y. Tao, and P. Kalnis, "On the Anonymization of Sparse High-Dimensional Data," Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE), pp. 715-724, 2008.

[9] R. J. Bayardo and R. Agrawal, "Data Privacy through Optimal k- Anonymization," in Proc. of ICDE, 2005, pp. 217–228.

[10] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-domain k-Anony Anonymity," in Proc. of ACM SIGMOD, 2005, pp. 49– 60.

[11] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional k-Anonymity," in Proc. of ICDE, 2006.

[12] Gabriel Ghinita, Member IEEE, Panos Kalnis, Yufei Tao," Anonymous Publication of Sensitive Transactional Data" in Proc. Of IEEE Transactions on Knowledge and Data Engineering February 2011 (vol. 23 no. 2) pp. 161-174.

[13] D.J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J.Y. Halpern, "Worst-Case Background Knowledge for Privacy- Preserving Data Publishing," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), pp. 126-135, 2007.

[14] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 139-150, 2006.

[15] Y. He and J. Naughton, "Anonymization of Set-Valued Data via Top-Down, Local Generalization," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 934-945, 2009.

[16] D. Kifer and J. Gehrke, "Injecting Utility into Anonymized Data Sets," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), pp. 217-228, 2006.

[17] T. Li and N. Li, "On the Tradeoff between Privacy and Utility in Data Publishing," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 517-526, 2009.

[18] Y. Xu, K. Wang, A.W.-C. Fu, and P.S. Yu, "Anonymizing Transaction Databases for Publication," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 767-775, 2008.

[19] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A.W.-C. Fu, "Utility- Based Anonymization Using Local Recoding," Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 785-790, 2006.

[20] C. Dwork, "Differential Privacy: A Survey of Results," Proc. Fifth Int'l Conf. Theory and Applications of Models of Computation (TAMC), pp. 1-19, 2008.

[21] C. C. Aggarwal and P. S. Yu. A condensation approach to privacy preserving data mining. In Proc. of the International Conference on Extending Database Technology (EDBT), pages 183–199, 2004.ory and Applications of Models of Computation (TAMC), pp. 1-19, 2008.

[22] C. C. Aggarwal and P. S. Yu. Privacy-Preserving Data Mining: Models and Algorithms. Springer, March 2008.

[23] R. Agrawal and R. Srikant. Privacy preserving data mining. In Proc.of ACM International Conference on Management of Data (SIGMOD), pages 439–450, Dallas, Texas, May 2000.

[24] L. Sweeney. k-Anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, 10(5):557–570, 2002.

[25] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey on recent developments. ACM Computing Surveys, 42(4), December 2010.

[26] J. Gehrke. Models and methods for privacy-preserving data publishing and analysis. In Tutorial at the 12th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Philadelphia, PA, August 2006.

[27] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," 1998. Technical Report, SRI-CSL-98-04, SRI International. 97

[28] Z. Yang, S. Zhong, and R. N.Wright. Anonymity-preserving data collection.In Proc. of the 11th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), pages 334–343, 2005.

[29] D. M. Carlisle, M. L. Rodrian, and C. L. Diamond. California inpatient data reporting manual, medical information reporting for california, 5th edition. Technical report, Office of Statewide Health Planning and Development, July 2007.

[30] X. Xiao and Y. Tao, "Anatomy: simple and effective privacy preservation," in Proceedings of the International Conference on Very Large Data Bases (VLDB), pp. 139–150, 2006.

[31] N. Koudas, D. Srivastava, T. Yu, and Q. Zhang, "Aggregate query answering on anonymized tables," in Proceedings of the International Conference on Data Engineering (ICDE), pp. 116–125, 2007.

Books:-

[32] Communications of the Association for Information Systems (Volume 8, 2002) 267-296.

[33] Data Mining: A Conceptual Overview by J. Jackson.

[34] TEXT MINING: CLASSIFICATION, CLUSTERING, AND APPLICATIONS By Ashok N. Srivastava and Mehran Sahami.

[35] INTRODUCTION TO PRIVACY-PRESERVING DATA PUBLISHING: CONCEPTS AND TECHNIQUES by Benjamin C. M. Fung, Ke Wang, Ada Wai-Chee Fu,and Philip S. Yu.

Websites:-

[36] Funk S**.** Netflix update: Try this at home [Online]. - 12 11, 2006. - http://sifter.org/~simon/journal/20061211.html.

[37] The Internet Movie Database. - 2007. - http://www.imdb.com/

[38] http://sites.google.com/site/litiancheng/

[39] http://surveillance.cancer.gov/joinpoint/aapc.html.

[40] http://www.trl.ibm.com/projects/security/ssp/main.html, June 2000

