ISSN:0975-9646

Data Leakage Detection with K-Anonymity Algorithm

Wakhare Yashwant R[#1], B. M. Patil[*2]

[#]*MBES's College of Engineering, Ambajogai,*
Dr. Babasaheb Ambedkar Marathwada University,
Aurangabad, India

***Abstract*—A data owner or a outsourcing company distributes the crucial information to a number of trusted agent or a organization (third parties). Few data can be leaked and found in non authorized place. The data owner will be finding out the agent who receives the actual data. Recently watermarking technique have been successfully used for copyright protection of multimedia data, the research of database watermarking scheme is still facing a lot of challenges due to the differences between the relational database and multimedia data. Some of the times watermarks can be destroyed if the data recipient is malicious. This paper focuses on detecting the distributor's crucial information that has been leaked by the agent, and it is possible to identify the agents that who leaks the data. "Guilt" probability model for data leakage is also available here. In few cases we can also inject "realistic but fake" object into the original data set to further improve our chances of detecting leaks and find out the guilty person. K-anonymity algorithm is used to create a sensitive data, so data set will be hidden and third parties will not be able to view the original data sets.**

***Keywords*—Data Leakage, K-anonymity, Fake Objects, Data Allocation, Guilt Model.**

I. INTRODUCTION

In the data mining the extraction of hidden, predictive information patterns from large data bases. It is helpful to identify the relevant and useful information from data bases. The overall goal of the data mining process is to extract information from a data set and transform it into understandable structure [14] [13]. Sometimes crucial information is leaked and found in non authorized place. For example, a college may have partnership with other colleges that require sharing the student data. Another enterprise may outsource its data processing, so data must be given to various other colleges. The owner of the data is called the distributor and the supposedly trusted third parties are called the agents. The aim is to detect when the distributors crucial data has been leaked by agents, and if possible to identify the agent that leaked the data [1] [2] [20]. A model for finding the guilty party is presented here. Also, an algorithm is presented for distributing objects to agents, in a way that improve the chances of identifying a leaker. The option of adding fake objects is considered to the distributed set. Such object do not correspond to real entities but appear realistic to agent. In a sense, the fake object act as a type of watermark for entire set, without modifying any individual members [11]. If it turns out that an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty. The optimization is considered in which leaked data is compared with original data and accordingly the third party who leaked the data is guessed. K-anonymity algorithm is proposed to create sensitive data, so the data will be hidden and third parties can not able to view the original data set.

This paper is organized as follows: In section II, Literature Review of paper. In section III, Explained Proposed System. In section IV, Results and Discussion and in section V and VI, future work and conclusion respectively.

II. LITERATURE REVIEW

Number of data leakage detection algorithm have been proposed in past, some are explaining as follows.

2003: Rights protection for relational data is handles data security through watermarking in the framework of numeric relational data and instead of primary key it uses the most significantly bits of the normalized data set. R.Sion, M. Atallah, and S. Prabhakar. Have proposes a watermark embedding algorithm such that it consist of sorting, partitioning used for marker location and bit embedding watermark bits are embedded in the numbers set. So as to provide a right protection to the data that are stored into it the relational databases [3].

2002: Generalization and suppression techniques to safeguard the data from the data distributors using K-anonymity privacy protection. The data in the system is analyzed for generalization, like replacing or recording a value with a less specific but semantically consistent values and suppression involves not releasing a value at all. It achieves that the released records adhere to K-anonymity, which means each released record has at least (k-1) other records. In the release whose values are indistinct over those fields that appear in external data [10].

2002: Watermarking the relational databases suggested that watermark can be applied to any database relation having attributes which are such that changes in a few of their values do not affect the application. R. Agrawal and J.Kiernan enunciates the need for watermarking databases relations to detect their piracy, identify the unique characteristics of relational data which pose new challenges for watermarking, and provide desirable properties of a watermarking system for relational data [4].

2003: In a warehousing environment, the data lineage problem is that of tracing warehouse \data item back to the original source item from which they were derived. Y. Cui and J. Widom formally defines the lineage tracing problem in the presence of general data warehouse transformation, and they present algorithm for lineage tracing in this environment [5].

III. Proposed System

In the proposed system of "Data Leakage Detection with K-anonymity Algorithm" is used to remove the drawback of the watermarking technique and the creating fake objects, allocating data, using guilt probability model to find the guilty agent and creating sensitive data by using k-anonymity algorithm.

A. *Fake Objects*

Our use of fake objects is inspired by the use of "trace" records in mailing lists. The distributors may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. However, fake objects may impact the correctness of what agent do, so they may not always be allowable. In our case, we are perturbing the set of distributor objects by adding fake elements. In some applications, fake objects may cause fewer problems that perturbing real object. The fake objects created can be two types, explicit fake tuples or sample fake tuples.

B. *Data Allocation Problem*

Here the main focus of our paper is the data allocation problem, how can the distributor "intelligently" give data to agents in order to improve the chances of detecting a guilty agent? As illustrated in below figure, there are four instances of this problem we address, depending on the type of data requests made by agents and whether "fake objects" are allowed.

As shown in below figure, we represent our four problem instances are, explicit request with fake tuples, explicit request without fake tuples, sample request with fake tuples, and sample request without fake tuples.

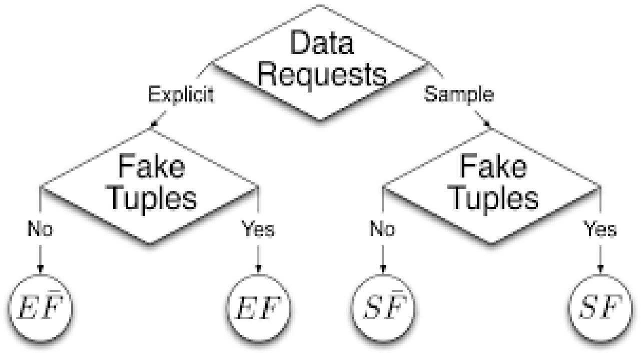

Fig. 1 Leakage problem instances

C. *Allocation Strategies*

There are two types of allocation strategies.

I) Explicit data request

II) Sample data request

I. *Explicit data request*

In case of explicit data request with fake not allowed, the distributor is not allowed to add fake object to the distributed data. So data allocation is fully defined by the agent's data request.

In case of explicit data request with fake allowed, the distributor cannot remove or alter the request R from the agent. However distributor can add the fake object. In algorithm for data allocation for explicit request, the input to this is a set of requests R_1, R_2,....R_n from n agents and different conditions for requests. The e-optimal algorithm finds the agents that are eligible to receiving fake object. Then create one fake object in iteration and allocate it to the agent selected. The e-optimal algorithm minimizes every term of the objective summation by adding maximum number b_i of fake objects to every set R_i yielding optimal solution.

TABLE I
EXPLICIT DATA REQUEST ALGORITHM [1]

```
Input: $R_1,..., R_n, cond_1,..., cond_n, b_1,...,b_n, B$
Output: $R_1,..., R_n, F_1,...,F_n$
$R \leftarrow \Phi$
For $i = 1,\dots, n$ do
    If $b_i > 0$ then
        $R \leftarrow R \cup \{i\}$
    $F_i \leftarrow \Phi$
While $B > 0$ do
    $i \leftarrow$ SELECTAGENT($R, R_1,\dots,R_n$)
    $f \leftarrow$ CREATEFAKEOBJECT($R_i, F_i, cond_i$)
    $R_i \leftarrow R_i \cup \{f\}$
    $F_i \leftarrow \cup \{f\}$
    $b_i \leftarrow b_i - 1$
    If $b_i = 0$ then
        $R \leftarrow R \setminus \{R_i\}$
    $B \leftarrow B - 1$
```

Algorithm makes a greedy choice by selecting the agent that will yield the greatest improvement in the sum-objective

TABLE II
AGENT SELECTION FOR E-RANDOM ALGORITHM [2]

```
function SELECTAGENT($R, R_1,\dots,R_n$)
    $i \leftarrow$ select at random an agent from R
return i
```

Explicit data request and e-random algorithm represent the stratergy for randomly allocating fake objects. Algorithm 1 is a general "driver" that will be used by other strategies, while e-random algorithm performs the random selection [14]. The combination of explicit data request and e-random algorithm denotes with as e-random. The use of e-random is baseline in comparison with other algorithm for explicit data request.

TABLE III
AGENT SELECTION FOR E-OPTIMAL ALGORITHM [3]

```
function SELECTAGENT(R, R_1,…,R_n)
    i ← $\underset{i:R_i \in R}{\arg\max}\left(\frac{1}{|R_i|}-\frac{1}{|R_i|+1}\right)\sum_j |R_i \cap R_j|$
return i
```

Algorithm 3 makes a greedy choice by selecting the agent that will yield the greatest improvement in the sum-objective. The cost of this greedy choice in $O(n^2)$ in every iteration. The overall running time of e-optimal is $O(n+n^2B)$.

II. Sample data request

The sample data request defines the object of S. It is used to compute the overall system reliability, while the probability is used to identify the guessing agents that have been leaked the information. These probabilities are estimated based on the experiments. Similarly, the probabilities are usually conservative estimates rather than exact numbers. While allocating data to the agents, that constraints is to satisfy the agent request by providing the entire request that is available and the objective is to detect the agent, who leaks the data.

TABLE IV
SAMPLE DATA REQUEST ALGORITHM [4]

```
Input: $m_1,..., m_n, |T|$
Output: $R_1,..., R_n,$
a ← $0_{|T|}$
$R_1$ ← Φ,…, $R_n$ ← Φ
remaining ← $\sum_{i=1}^{n} m_i$
while remaining > 0 do
    for all I = 1,…,n : $|R_i| < m_i$ do
        k ← SELECTOBJECT(I, $R_i$)
        $R_i$ ← $R_i$ U { $t_k$ }
        a[k] ← a[k] + 1
        remaining ← remaining - 1
```

Sample data request algorithm defines the sample data request [12]: the distributor can minimize both objectives by allocating distinct sets to all agents. Such an optimal allocation is possible, since agents request in total fewer objects than the distributor has. The distributor can achieve such an allocation by using sample data request algorithm. It denotes the resulting algorithm as s-overlap. It does not minimize sum-objective. However, s-overlap does minimize the sum of overlaps. In this, sample data request algorithm provides an agent U_1 with an object that has been given to the smallest number of agents. Every agent will receive a data set with objects that no other agent.

D. K-anonymity algorithm

K-anonymity provides simple and effective approaches to protect private information of individuals via only releasing k anonymous views of a data set. An anonymised data set contains multiple fields that can be used to identify someone (e.g. age, sex, location). Thus, the k-anonymity model has gained increasing popularity. If a table satisfies k-anonymity for some value k, then anyone who knows only the quasi-identifier values of one individual cannot identify the record. Quasi-identifier represents the primary type of disclosure risk that needs to be focused on its identity disclosure. An underlying assumption for this type of risk is that there is two pieces of information: (a) the actual data set that has been disclosed and (b) some background information about one or more people in this data set. The background information is described by a set of variables. These variables are called quasi-identifiers. For example each row in the table cannot be distinguished from at least other k-1 rows by only looking a set of attributes; this table is k-anonymised on these attributes. This can uniquely identify an individual directly.

TABLE V
K-ANONYMITY ALGORITHM [6]

```
Fully generalize all tuples such that all tuples are equal
let P be a set containing all these generalized tuples
S←{P};0←θ
repeat
S←θ
for all P€ S do
specialize all tuple in P one level down in the generalization hierarchy such that a number of specialized child nodes are formed
unspecialized the nodes which do not satisfy (α,k)-anonymity by moving the tuples back to the parent node
if the parent P does not satisfy (α,k)-anonymity then
unspecialized some tuples in the remaining child nodes so that the parent P satisfies (α,k)-anonymity
for all non empty branches B of P, do S ← S U{B}
S← S
if P is non empty then 0← 0 U{P}
until S = ϴ
return 0
```

This method includes the following steps:

1. Construct and K-anonymous table T from the given raw table (which will be described in K-anonymity Algorithm), and assign each equivalence class in the resulting table a class ID.
2. Add a column for the class ID of the equivalence class in the original raw table, such that, for each tuple, the class ID is the ID of the equivalence class that the tuple belongs in T. Call this new table the Temp table. Hence the Temp table contains the raw table plus one extra column.
3. The Temp table on the QID attributes and the Class ID column.
4. Project the Temp table on the sensitive attributes and the Class ID column. This results in the SS table. The following attributes defines as a sensitive data

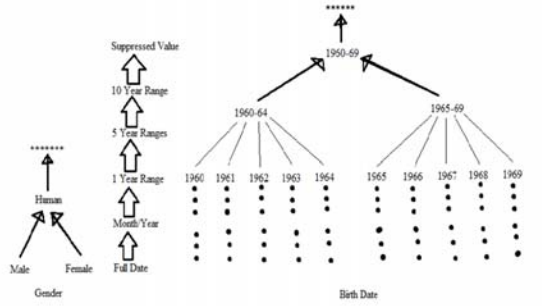

Fig. 2 Sensitive attributes using K-anonymity

Figure 2 represents the sensitive attributes using K-anonymity. While k-anonymity algorithm is applied after the data is sensitive and the sensitive data also hidden. In this K-anonymity provides privacy protection by guaranteeing that each record relates to at least k individuals even if the released records are directly linked to external information. It provides a formal presentation of achieving k-anonymity using generalization and suppression.

The main focus of this research is the data allocation problem. The distributor "intelligently" gives data to agents in order to improve the chances of detecting a guilty agent. Data leakers are called guilty agents. Fake objects are generated by the distributor in order to increase the chances of detecting agents that leak data. The distributor adds fake objects and creative sensitive data by using k-anonymity algorithm applied to the distributed data, in order to improve the effectiveness of detecting guilty agents.

IV.Results And Discussion

In data leakage detection, k-anonymity algorithm is used to detect the leakages. The following graphs represent the probability (of finding the guilty agent) comparison between the different agents and the efficiency comparison of existing system and proposed system. The result of these experiments is discussed.

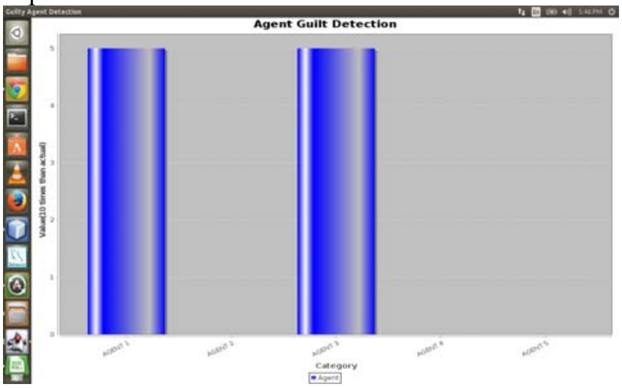

Fig. 3 Diagram of finding the guilty agent

Figure 3 represents the comparison graph between the two agents. In this graph agent 1 is having the probability value is 5 and the agent 2 is also having the probability value 5. So the existing algorithm is having less efficiency to find out the exact guilty agent as compare to k-anonymity algorithm. Efficiency comparison graph is as shown below.

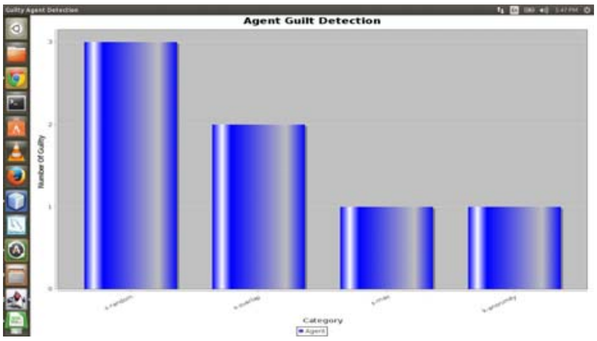

Fig. 4 Efficiency comparison between the existing and proposed algorithm.

Figure 4 represent the efficiency comparison graph between the existing and proposed algorithm(k-anonymity). In this graph existing algorithms(s-random, s-overlap, s-max) are having the less efficiency to find out the guilty agent. And the k-anonymity algorithm is having the high efficiency to find out the exact guilty agent.

This proposed method is to simulate the data leakage problems to evaluate their performance. The goal of these experiments was to seen whether the fake objects in distributed data sets yield significant improvement in this chance of detecting guilty agents. If it turns out an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty. Next step is to evaluate the e-optimal algorithm relative to a random allocation. It focuses on a few objects that are shared among multiple agents. Since object sharing makes it difficult to distinguish a guilty from non-guilty agents. More objects to distribute with objects shared among fewer agents are obviously easier to handle.

V.Future Work

The k-anonymity protection model is presented here to explore related attacks and provide ways in which these attacks can be detected. In future, t-closeness can be used to overcome k-anonymity background knowledge attacks that enable new kinds of privacy threats on sequential data releases.

VI.Conclusion

In doing business there would be no need to handover sensitive data to agents that may unknowingly or maliciously leak it. And even if we had to handover sensitive data, in a perfect world we could trace its origins with absolute certainty. However, in many cases we must indeed work with agents that may not be 100% trusted, and we may not be certain if a leaked object came from an agent or from some other source. In spite of these difficulties we have shown it is possible to assess the likelihood that an agent is responsible for a leak, based on the overlap of his data with the leaked data and the data of other agents, and based on the probability that onjects can be "guessed" by other means.

References

[1] Chih-Hua Tai, Philip S. and De-Nian Yang. Privacy Preserving Social Network Publication Against Friendship Attack, University of Illinois at Chicago.

[2] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, Nikos Mamoulis, Fast Data Anonymization with Low Information Loss, 2007.

[3] R.Sion, M. Atallah, and S.Prabhakar, " Right Protection for R elational D atabase," proc. ACM SIGMOID, pp. 98-109, 2003.

[4] R. Agrawal and J. Kiernan,"Watermarking Relational Databases", proc. 28th international conference very large data bases (VLDB'02), VLDB endowment, pp. 155-166, 2002.

[5] Y.Cui and J. Widom, "Lineage Tracing for General Data Warehouse Transformations", The VLDB J. vol.12, pp. 41-58, 2003.

[6] Latanya Sweeney School of Computer Science Achieving K-Anonymity Privacy Protection Using Generalization and Suppression, Carnegie. Mellon University, Pittsburgh Pennsylvania, USA, 2007.

[7] Latanya Sweeney, Guaranteeing Anonymity when Sharing Medical Data, the Datafly System, Massachusetts Institute of Technology Cambridge.

[8] Latanya Sweeney, k-Anonymity: A model for protecting privacy, Carnegie Mellon University, may 2007.

[9] Latanya Sweeney, towards the Optimal Suppression of Details when Disclosing Medical Data, the Use of Sub-combination Analysis, Massachusetts Institute of Technology.

[10] L. Sweeney, Achieving k-anonymity privacy protection using generalization and suppression. International Journal on Uncertainty, Fuzziness and Knowledge-based System, 10(5), 2002: 571-588.

[11] Mohammed Shehab, Elisa Bertino and Arif Ghafoor, Water marking For Relational Data Base By Using Threshold Generator IEEE Transaction On Knowledge and Data Engineering, Vol.22 No 3 2011.

[12] Naresh Bollam and Mr. V.Malsoru Review on data leakage detection, International Journal of Engineering Research and Application (IJERA) 2011.

[13] N.Sandhya, G.Haricharan Sharma and K.Bhima. Exerting Modern Techniques for Data Leakage Problems. Detect, International Journal of Electronics Communication and Computer Engineering 2012

[14] P.Papadimitriou and H. Garcia-Milina Data Leakage detection. Technical report, Stanford University, 2010.

[15] Pradnya B. Rane and B. B. Meshram. Application Level and Database Security for E-Commerce Application, (0975-8887) march 2012.

[16] Radu Sion, Mikhail Atallah, Fellow, IEEE, and Sunil Prabhakar, Rights Protection for Relational Data, Vol. 16, No.6, June 2007

[17] Rakesh Agrawal, Jerry Kiernan, Watermarking Relational Databases, IBM Almaden Research center, CA 95120

[18] Robert Ikeda and Jennifer Widom, Data Lineage: A survey, Stanford university.

[19] P.Buneman and W. C. Tan,"Provenance in Databases," proc. ACM SIGMOID, PP. 1171-1173, 2007.

[20] Thomas H. Hinke, Inference Aggregation Detection In Database Management System, 2008.