IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 12, Issue 3 (Jul. - Aug. 2013), PP 119-129
www.iosrjournals.org

A Robust Approach for Detecting Data Leakage and Data Leaker in Organizations

B. Sruthi Patel[1], M. L. Prasanthi[2], Chinni. Jayachandra[3]
[1]*M.Tech (CSE), VCE, Hyderabad India.*
[2]*Assoc.Professor.inCSE, VCE, Hyderabad India.*
[3]*M.Tech (CSE), VCE, Hyderabad India.*

Abstract: *In organizations sensitive data transaction is having less security because there may be a misusability of data from one to other from past years. In previous methods they calculated how much data is leaked but didn't find who is leaked. In this paper, we find leakage and leaker. A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). Some of the data has leaked and found in an unauthorized place (e.g., on the web or somebody's laptop). The distributor should assess the likelihood of the leaked data came from one or more agents, as opposed to having independently gathered by others. We propose data allocation strategies (across the agents) that improve the probability of identifying leakages. These methods don't rely on alterations of the released data (e.g., watermarks). In some cases we can also inject "realistic but fake" data records to further improve our chances of detecting leakage and identifying the guilty party.*

Key Words: *Data Leakage, fraud detection, agent guilt model, Explicit Random, Sample Random.*

Submitted date 21 June 2013 **Accepted Date: 26 June 2013**

I. Introduction

In the course of doing a business, sometimes sensitive data must be handed over to supposedly trusted agents. For example, a company may have partnerships with the other companies that require sharing customer data. Another enterprise might outsource its data processing, so that the data must be given to various other companies. Our aim is to detect when the distributor's sensitive data have been leaked by agents, and if possible to identify the particular agent that leaked the data. Perturbation is most useful technique where the data has modified and made "less sensitive" before being handed to agents For example, one can replace the exact values by ranges, or one can add the random noise to certain attributes. Traditionally, watermarking is used to handle the leakage detection. We annunciate the need for watermarking database relations to deter their piracy, and identify the unique characteristics of relational data which pose new challenges for watermarking, and provide desirable properties of watermarking system for relational data. A watermark can be applied to any of the database relation having attributes which are such that changes in a few of their values do not affect the applications. Watermarking means a unique code is embedded in each distributed copy if that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious.

SENSITIVE information such as customer or patient data and business secrets constitute the main assets of an organization. Such information is essential for the organization's employees, subcontractors, or partners to perform their tasks. Conversely, limiting access to the information in the interests of preserving secrecy might damage their ability to implement the actions that can best serve the organization. Thus, data leakage and data misuse detection mechanisms are essential in identifying malicious insiders. The task of detecting malicious insiders is very challenging as the methods of deception become more and more sophisticated. According to the 2010 Cyber Security Watch Survey 26 percent of the cyber-security events, recorded in a 12-month period, were caused by insiders. These insiders were the most damaging with 43 percent of the respondents reporting that their organization suffered data loss. Of the attacks, 16 percent were caused by theft of sensitive data and 15 percent by exposure of confidential data. In recent days, leakage detection is prevented by watermarking, e.g., a unique code is embedded in each circulated copy. If that copy is discovered in the other hands or illegal parties, the leaker can be identified with the watermarking. Watermarks are helpful in some cases, but again, involve some modification of the original data. In addition, watermarks can sometimes be cracked if the data recipient is mean. Amir Harel proposed M-Score [1] is to find the how much data is leaked in organization. Jagtapn.p Data leakage detection system is to improve the efficiency of finding data leakage [2], [3].

In this paper, we study techniques for detecting leakage of a set of objects or records. Specifically, we study the following scenario: After giving a set of objects to agents, the distributor discovers some of those same objects in an unauthorized place.At this point, the distributor can assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means data, he may stop doing business with him, or may initiate legal proceedings. In this paper, we develop a model for assessing the "guilt"of agents. We also present algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker. Finally, we also consider the option of adding"fake" objects to the distributed set. Such objects do not correspond to real entities but appear realistic to the agents.In a sense, the fake objects act as a type of watermark for the entire set, without modifying any individual members.

II. Data Allocation Problem

In organization database it contains all the information about the partners, customers and agents as employee etc. It contains sensitive which can be found in an unauthorized place as on someone's laptop, desktop of other person of some other organization, as the data have to protect from fraud (leaker). As the agent working within the organization will cause fraud by leaking the data to third parties. So, we have to protect our data from leakage, this task became more important in these days. In previous system we have find only the probability for leaked data as calculating the score given to the data records as M-Score but haven't find the leaked data and the fraud leaker, who cause leakage, but in this proposed system we can find the leaker and also the leaked data with the probability value given to the leaked data as how much the organization get loss on the leaked data.

Data leakage detection system Architecture will have the description about the detail process of Data Distributor task and agent's task and how the Distributor will find the fraud agent and the probability value for the leaked data and how the agent will leak the data to the third party. As Distributor will plays an important role as he have to give security to the organization's data. Distributor will have the agent details, if the agent is a new one he has to get register and get the username and password for login process.

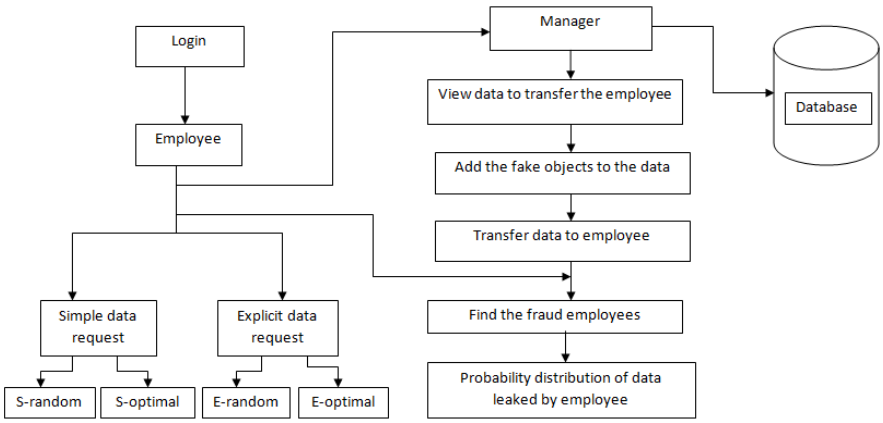

Fig 1. Architecture For Data Leakage And Finding Data Leaker System.

Agent will login using username and password given to him. Agent will check the records details and do the work which was assign to him. If the agent wants to leak the organization data, then he will send the sensitive records to third parties and continues his work to get more records. As the data has leaked and organization don't know who leaked data as they don't have the details' about leakage, as every agent will tells that he is not fraud as he have not leaked the data to third party. As, the Distributor also not know about the leakage as he won't point out any agent who works within that organization. Distributor will have the details of what the agent is actually doing with the data which is exposed to him while processing his task. The insider will cause for the leakage in most of the scenario. So, Distributor has to take care of data and also give response to agent when the agent send request to access data to do their task. And Distributor can change the original sensitive records data as less sensitive by adding fake record data which will look like realistic data records and the sensitive data will become less sensitive. As the agent send request to the Distributor to get the record details to do the task then the Distributor will add fake record to the original record based on the access request given to him by the agent. Distributor will observe the behavior of the agent as he is trying to access the sensitive data which he has no access permission the he will add fake record and forward to agent. As agent have no idea of adding fake object to the original record and he will send them to third party. Distributor only knows which fake object record has added to which original data and forwarded to which agent. Distributor will add one or more fake data to the original data based on sensitive data, and decreases its sensitivity.

Agent will transfer the data to third party and he tells that he has not leaked the data then the Distributor will catch the leaker from the number of agents employees based on the fake data present in the transfer data. Finally the fraud agent will catch and get punished based on his leakage. Here Distributor will calculate the probability of leakage and the score as how much the organization has lost because of that leakage. M-score can be given as how many records have been leaked out of original records.
Distributor will find the guilt agent based on fake records and the request send by the agents to the Distributor. We are using E-random and S-random. E-Random is used to find out the Agent who leaked the data and S-Random is used to find out the Leakage Data. E-optimal and S-optimal is used to get the optimization about the addition of fake objects.

2.1. Fake Objects:

Fake objects are objects generated by the distributor in order to increase the chances of detecting agents that leak data. The distributor may be adding fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. Our use of fake objects is inspired by the use of "trace" records in mailing lists. The idea of perturbing data to detect leakage is not new, e.g., [1]. However, in most cases, individual objects are Perturbed, e.g., by adding random noise to sensitive salaries, or adding fake elements.

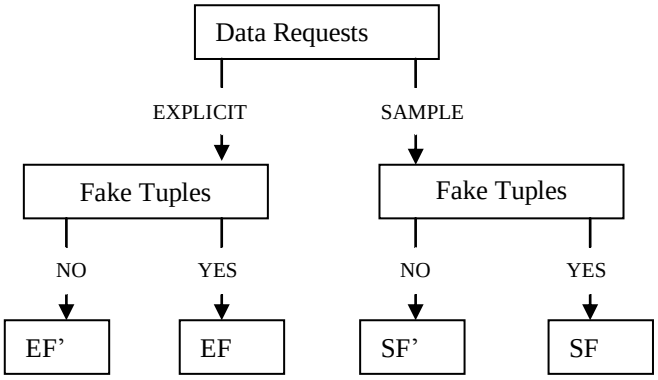

Fig 2. Leakage Problems Instances.

In some applications, fake objects may cause fewer problems that perturbing real objects creation. The creation of fake but real-looking objects is a nontrivial problem whose thorough investigation is beyond the scope of this paper. Here, we model the creation of a fake object for agent Ui as a black box function CREATEFAKEOBJECT $(R_i, F_i, condi)$ that takes as input the set of all objects R_i, the subset of fake objects F_i that U_i has received so far, and condi, and returns a new fake object.This function needs condi to produce a valid object that satisfies U_i's condition. Set R_i is needed as input so that the created fake object is not only valid but also indistinguishable from other real objects.

Although we do not deal with the implementation of CREATEFAKEOBJECT ( ), we note that there are two main design options. The function can either produce a fake object on demand every time it is called or it can return an appropriate object from a pool of objects created in advance. We are using the following strategies to add the fake object to finding guilty agent.

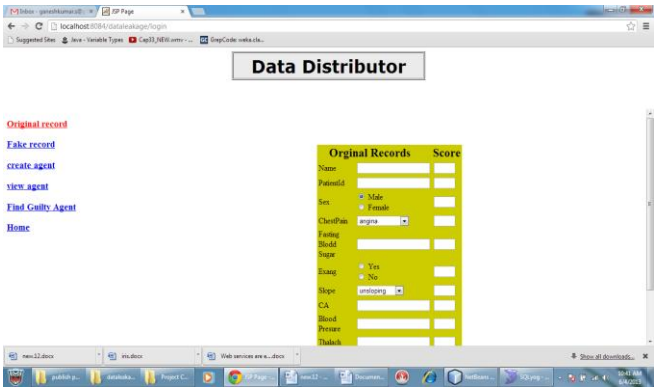

Fig3. Adding the Original records

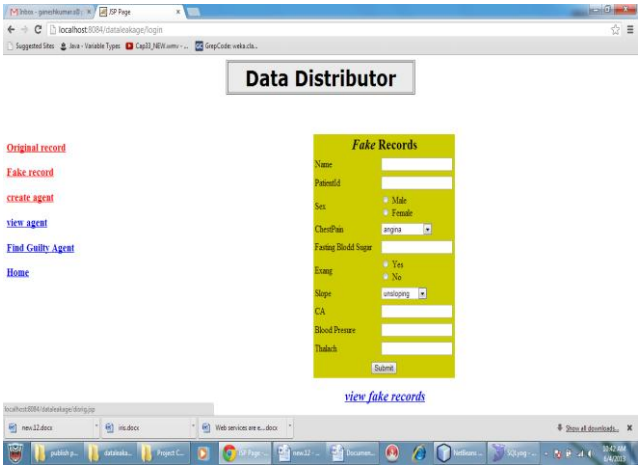

Fig 4. Adding the fake objects

2.2. Optimization Problem:

The Optimization Module is the distributor's data allocation to agents has one constraint and one objective. The distributor's constraint is to satisfy agents' requests, by providing them with the number of objects they request or with all available objects that satisfy their conditions. His objective is to be able to detect an agent who leaks any portion of his data.We consider the constraint as strict. The distributor may not deny serving an agent request and may not provide agents with different perturbed versions of the same objects as in [1]. We consider fake object distribution as the only possible constraint relaxation. Our detection objective is ideal and intractable.

Detection would be assured only if the distributor gave no data object to any agent. We use instead the following objective: maximize the chances of detecting a guilty agent that leaks all his data objects. We now introduce some notation to state formally the distributor's objective. Recall that $\Pr\left\{G_j \middle/ S = R_i\right\}$ or simply $\Pr\left\{G_j \middle/ R_i\right\}$ is the probability that agent U_j is guilty if the distributor discovers a leaked table S that contains all R_i objects. We define the difference functions $\Delta(\mathrm{i},\mathrm{j})$ as

$$\Delta(\mathrm{i},\mathrm{j}) = \Pr\left\{G_i \middle/ R_i\right\} - \Pr\left\{G_j \middle/ R_i\right\} \quad i, j = 1,\ldots,n \quad \ldots\ldots(1)$$

Problem Definition. Let the distributor have data requests from n agents. The distributor wants to give tables $R_1,\ldots,R_n$ to agents $U_1,\ldots,U_n$, respectively, so that . He satisfies agents' requests, and he maximizes the guilt probability differences $\Delta(I,j)(i,j)$ for all $i, j = 1, i \neq j$. Assuming that the R_i sets satisfy the agents' requests, we can express the problem as a multi criterion optimization problem:

maximize $(\ldots,\Delta(i,j),\ldots)$ $i \neq j$ .....(2),

If the optimization problem has an optimal solution, it means that there exists an allocation $D^* = \left(R_1^*,\ldots,R_n^*\right)$such that any other feasible allocation yields $D^* = \left(R_1^*,\ldots,R_n^*\right)$ yields $\Delta(i,j) \geq \Delta^*(i,j)$ for all $i; j$. This means that allocation T_j^* allows the distributor to discern any guilty agent with higher confidence than any other allocation, since it maximizes the probability $\Pr\left\{G_i \middle/ R_i\right\}$ with respect to any other probability $\Pr\left\{G_i \middle/ R_j\right\}$ with $j \neq i$. Even if there is no optimal allocation D^*, a multi criterion problem has Pareto optimal allocations.

2.3. Objective Approximation

We can approximate the objective of (2) with (3) that does not depend on agents' guilt probabilities, and therefore, on

$$p: Maximise\left(..., \left| R_i \cap R_j \right| \Big/ \left| R_i \right|, ... \right) i \neq j \quad .....(3)$$

This approximation is valid if minimizing the relative overlap $\left| R_i \cap R_j \right| \Big/ R_i$ maximizes $\Delta(i, j)$.

Therefore, we can scalarize the problem objective by assigning the same weights to all vector objectives.

$$\text{Maximize} \quad \sum i = 1 \, 1 \Big/ \left| R_i \right| \sum j = 1 \left| R_i \cap R_j \right| \quad .....(4a)$$

$$(over R_1 ... R_n)$$

$$\text{Maximize} \quad \max \left| R_i \cap R_j \right| \Big/ R_i \quad .....(4b)$$

$$(over R_1 ... R_n)$$

Both scalar optimization problems yield the optimal solution of the problem of (3), if such solution exists. If there is no global optimal solution, the sum-objective yields the Pareto optimal solution that allows the distributor to detect the guilty agent, on average (over all different agents), with higher confidence than any other distribution. The max objective yields the solution that guarantees that the distributor will detect the guilty agent with certain confidence in the worst case. Such guarantee may adversely impact the average performance of the distribution.

III. Allocation Strategies

The main focus of our project is the data allocation problem as how can the distributor "intelligently" give data to agents in order to improve the chances of detecting a guilty agent.

3.1. Explicit Data Requests

In problems of class EF , the distributor is not allowed to add fake objects to the distributed data. So, the data allocation is fully defined by the agents' data requests. Therefore, there is nothing to optimize. In EF problems, objective values are initialized by agents' data requests. Say, for example, that $T = \{t_1, t_2\}$ and there are two agents with explicit data requests such that $R_1 = \{t_1, t_2\}$ and $R_2 = \{t_1\}$. The value of the sum objective is in this case $\sum i = 1 \, 1 \Big/ \left| R_i \right| \sum j = 1 \left| R_i \cap R_j \right| = 1/2 + 1/1 = 1.5$.

The distributor cannot remove or alter the R_1 or R_2 data to decrease the overlap $R_1 \cap R_2$. However, say that the distributor can create one fake object $(B = 1)$ and both agents can receive one fake object. In this case, the distributor can add one fake object to either R_1 or R_2 to increase the corresponding denominator of the summation term. Assume that the distributor creates a fake object f and he gives it to agent R_1. Agent U1 has now $R_1 = \{t_1, t_2, f\}$ and $F_1 = \{f\}$ and the value of the sum-objective decreases to $1/3 + 1/1 = 1.33 < 1.5$.

*Algorithm*1.*Allocation for* Explicit *Data Requests*(EF)

Input : $\mathrm{R}_1....\mathrm{R}_n, \mathrm{cond}_1;....;cond_n, \mathrm{b}_1,...,\mathrm{b}_n, \mathrm{B}$

Output : $R_1....R_n, F_1...F_n$

1: $R \leftarrow$ *Agents that can receive fake objects*

2: *for* $i = 1....n$ *do*

3: *if* $b_i > 0$ *then*

4: $R \leftarrow R \cup \{i\}$

5: $F_i \leftarrow \phi;$

6: *while* $B > 0$ *do*

7: $i \leftarrow SELECTAGENT(R, R_1....R_n)$

8: $f \leftarrow CREATEFAKEOBJECT(R_i, F_i, cond_i)$

9: $R_i \leftarrow R_i \cup \{f\}$

10: $F_i \leftarrow F_i \cup \{f\}$

11: $b_i \leftarrow b_i\_1$

12: *if* $b_i = 0$ *then*

13: $R \leftarrow R \big/ \{R_i\}$

14: $B \leftarrow B\_1$

Algorithm 2.*Agentselection for* $e-random$

1: *function* $SELECTAGENT(R, R_1;...;R_n)$

2: i $\leftarrow$ *select at random an agent from R*

3: *return* i

In lines 1-5, Algorithm 1 finds agents that are eligible to receiving fake objects in $O(n)$ time. Then, in the main loop in lines 6-14, the algorithm creates one fake object to every iteration and also allocates it to random agent. The main loop takes $O(B)$ time. Hence, the running time of the algorithm is $O(n+B)$. If $B \geq \sum n, i = 1, b_i$ the algorithm minimizes every term of the objective summation by adding the maximum number bi of fake objects to every set Ri, yielding the optimal solution. Otherwise, if $B < E_i = 1$, b_i (as in our example where $B = 1 < b_1 + b_2 = 2$), the algorithm just selects at random the agents that are provided with fake objects. We return back to our example and see how the objective would change if the distributor adds fake object f to R_2 instead of R_1. In this case, the sum-objective would be $1/2 + 1/2 = 1 < 1.33$. The reason why we got a greater improvement is that the addition of a fake object to R_2 has greater impact on the corresponding summation terms, since

$$1/|R_1| - 1/|R_1| + 1 = 1/6 < 1/|R_2| - 1/|R_2| + 1 = 1/2.$$

The left-hand side of the inequality corresponds to the objective improvement after the addition of a fake object to R_1 and the right-hand side to R_2.

Algorithm 3. Agent Selection for e-optimal

1: $function(R, R_1;...;R_n)$

$$2: i \leftarrow \arg\max \left(1/|R_i'| - 1/|R_i'| + 1 \right) \sum |R_i' \cap R_j|$$

3: *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 is $o(n_2)$ for every iteration. The overall running time of e-optimal is $O(n+n2B)=O(n2B)$. Theorem 2 shows that this greedy approach finds an optimal distribution with respect to both optimization objectives defined in (4).
Theorem 2: Algorithm e-optimal yields an object allocation that minimizes both sum- and max-objective in problem instances of class EF .

3.2. Sample Data Requests

With sample data requests, each agent U_i may receive any T subset out of $(|T|)$ different ones. Hence, there are $i=1(|T|)$ different object allocations. In every allocation, the distributor can permute T objects and keep the same chances of guilty agent detection. The reason is that the guilt probability depends only on which agents have received the leaked objects and not on the identity of the leaked objects. Therefore from distributor's perspective, there are different allocations. The distributor's problem is to pick one out so that he optimizes his objective. We formulate the problem as a nonconvex QIP that is NP-hard.
Note that the distributor can increase the number of possible allocations by adding fake objects (and increasing $|T|$) but the problem is essentially the same. So, in the rest of this section, we will only deal with problems of class SF , but our algorithms are applicable to SF problems as well.

Fig 5. Requesting The Data

3.3. Random

An object allocation that satisfies requests and ignores the distributor's objective is to give each agent U_i a randomly selected subset of T of size mi. We denote this algorithm by S-random and we use it as our baseline. We present S-random in two parts: Algorithm 4 is a general allocation algorithm that is used by other algorithms in this section. In line 6 of Algorithm 4, there is a call to function SELECTOBJECT () whose implementation differentiates algorithms that rely on Algorithm 4. Algorithm 5 shows function SELECTOBJECT () for s-random.
Algorithm 5: Object selection for s-random
1: function SELECTOBJECT(i,R_i)
2: k ← selects at random an element from set
$\{K'/tk \notin Ri\}$
3: Return k.

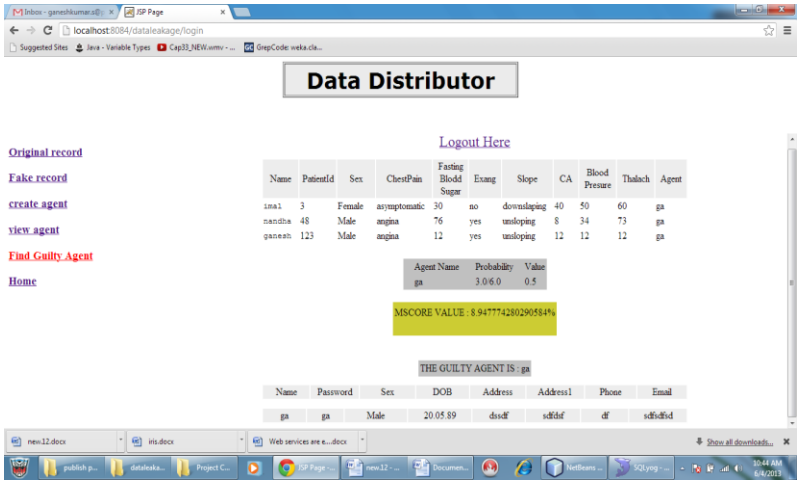

Fig 6. Finding Fake Employee By Calculating M-Score.

This page will displays the details of data records the agent has transfers to third-party and probability value and M-Score value if he tried to fraud with the details about records.

Algorithm For Find Guilt Agent

Step 1: Distributor selects the agents to send the data according to agent request.

Step 2: The distributor can create fake data and distribute with agent data or without fake data. Distributor is able to create more fake data; he could further improve the chance of finding guilt agent.

Step 3: Distributor checks the number of agents, who have already received data.

Step 4: Distributor chooses the remaining agents to send the data. Distributor can increase the number of possible allocations by adding fake data.

step 5: Estimate the probability value for guilt agent. To compute this probability, we need an estimate for the probability that values can be "guessed" by the target.

IV. Performance And Results

The performance of our approach is shown in below figures. Distributor creates the agent registration with relevant data in fig 8. Agent is having username and password to access data. Distributor send data to the agent with fake data as shown in fig 9. If agent farward the data to other person or third person then distributor will check his score fig 12. By obtaining the score we get how much data is leaked and using fake data find the data leaker in fig 13.

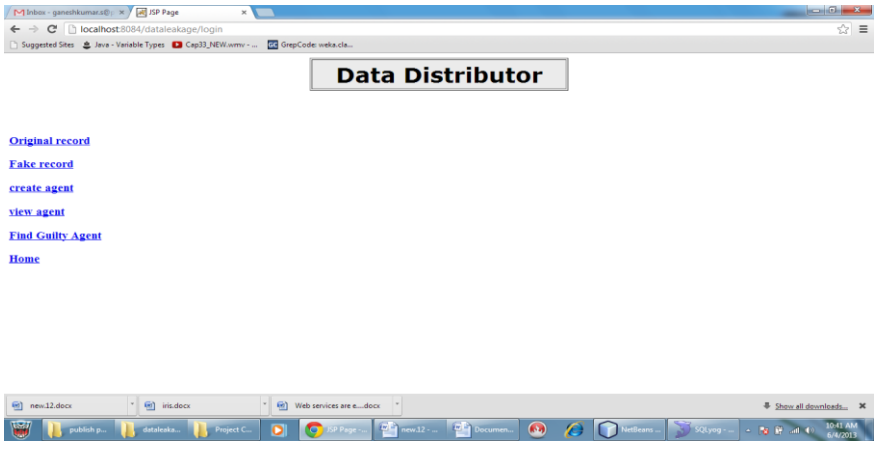

Fig 7.Data Distributor To Create New Agent And View Agent Details And Also He Will Find Guilty Agent Details With Probability Value And M-Score Value If He Leak The Data.

Fig 8. Creating agent by distributor.

Fig 9.Trnasfering Original record.

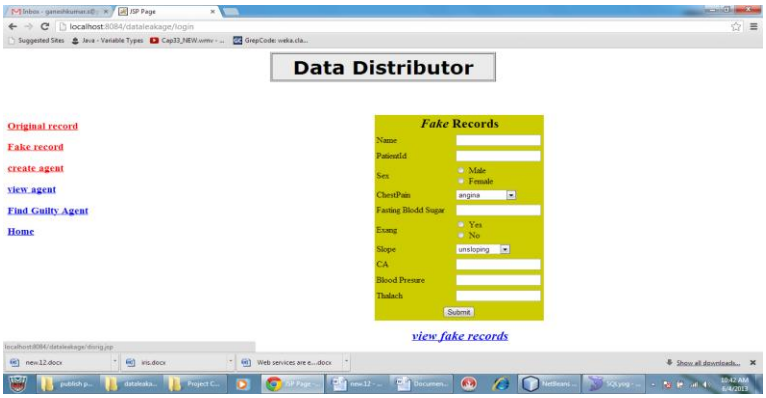

Fig 10.Distributor adding fake record with original data using explicit and Samples.

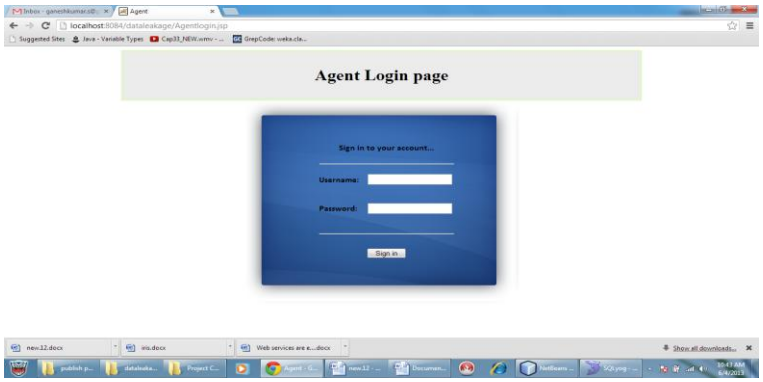

Fig 11. Agent (employee) login page,where agent will login with username and password for authentication process.

Fig 12. Display details about request type and also the details about parameters and the data transfers to third party.

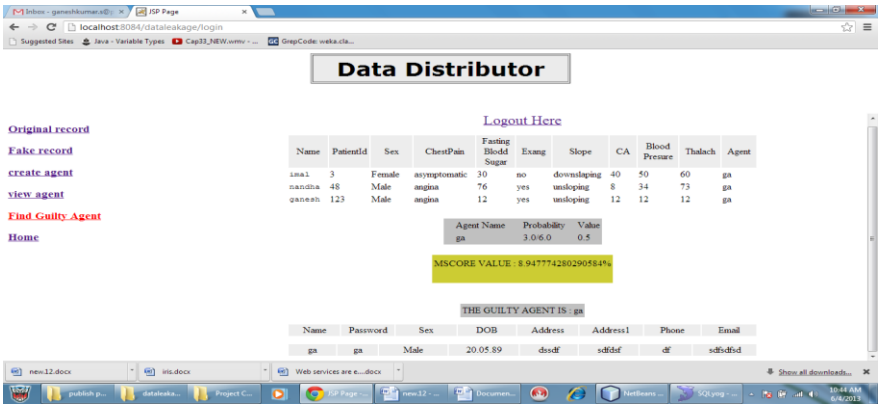

Fig 13. Displays The Details Of Data Records The Agent Has Transfers To Third-Party And Probability Value And M-Score Value If He Tried To Fraud With The Details About Records (Finding Leakage And Leaker).

V. Conclusions

In real world, the organizations are facing difficulties regarding malicious data leakage. A new concept of misuseability weight and discussed the importance of measuring the sensitivity level of the data that an insider is exposed to. There is no previously proposed method for estimating the potential harm that might be caused by leaked or misused data while considering important dimensions of the nature of the exposed data. Consequently, a new misuseability measure, the M-score, was proposed. In recent approaches we find how much data is leaked in the organization. By using our approach, find the employee who leaked the data using adding fake object to the original data. E-Random and S-Random are used to find the employee. This approach improves the efficiency for finding data leaker.

References

[1] Amir Harel, AsafShabtai, LiorRokach, and Yuval Elovici, "M-Score: Misusability weight measure". IEEE ransactions, june 2012.
[2] N.Sandhya , K. Bhima , G. Haricharan Sharma," Exerting Modern Techniques for Data Leakage Problems Detect" International Journal of Electronics Communication and Computer Engineering- 2012, Volume 3, Issue 1.
[3] ArchanaVaidya, PrakashLahange, Kiran More, ShefaliKachroo&NiveditaPandey. "DATA LEAKAGE DETECTION ".IJAET, 2012.
[4] A.subbiah and D.M.Blough.An Approach for fault tolerant and securedata storage in collaborative Work environments.
[5] M. Atallah and s.Wagstaff, "Watermarking with quadratic residues". In proctor IS&T/SPIE Conference on Security and Watermarking of Multimedia Contents, January 1999.
[6] P. Buneman and W.C. Tan, "Provenance in Databases", Proc. ACM SIGMOD, pp. 1171-1173, 2007.
[7] S.Katzenbeisser and F.A.peticolas , editors. Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, 2000.
[8] R. Agrawal and J. Kiernan, "Watermarking Relational Databases,"Proc. 28th Int'l Conf. Very Large Data Bases (VLDB '02), VLDB Endowment, pp. 155-166, 2002.
[9] Y. Cui and J. Widom, "Lineage Tracing for General Data Warehouse Transformations," The VLDB J., vol. 12, pp. 41-58,2003.
[10] F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video," Signal Processing, vol. 66, no. 3, pp. 283-301,1998.
[11] S. Jajodia, P. Samarati, M.L. Sapino, and V.S. Subrahmanian,"Flexible Support for Multiple Access Control Policies," ACM Trans. Database Systems, vol. 26, no. 2, pp. 214-260, 2001.
[12] Y. Li, V. Swarup, and S. Jajodia, "Fingerprinting Relational Databases: Schemes and Specialties," IEEE Trans. Dependable and Secure Computing, vol. 2, no. 1, pp. 34-45, Jan.-Mar. 2005.
[13] B. Mungamuru and H. Garcia-Molina, "Privacy, Preservation and Performance: The 3 P's of Distributed Data Management," technical report, Stanford Univ., 2008.

[14] V.N. Murty, "Counting the Integer Solutions of a Linear Equation with Unit Coefficients," Math. Magazine, vol. 54, no. 2, pp. 79-81,1981.
[15] S.U. Nabar, B. Marthi, K. Kenthapadi, N. Mishra, and R. Motwani, "Towards Robustness in Query Auditing," Proc. 32nd Int'l Conf. Very Large Data Bases (VLDB '06), VLDB Endowment, pp. 151-162,2006.
[16] P. Papadimitriou and H. Garcia-Molina, "Data Leakage Detection,"technical report, Stanford Univ., 2008.
[17] P.M. Pardalos and S.A. Vavasis, "Quadratic Programming with One Negative Eigenvalue Is NP-Hard," J. Global Optimization,vol. 1, no. 1, pp. 15-22, 1991.
[18] J.J.K.O. Ruanaidh, W.J. Dowling, and F.M. Boland, "WatermarkingDigital Images for Copyright Protection," IEE Proc. Vision, Signal and Image Processing, vol. 143, no. 4, pp. 250-256, 1996.
[19] R. Sion, M. Atallah, and S. Prabhakar, "Rights Protection for Relational Data," Proc. ACM SIGMOD, pp. 98-109, 2003.
[20] L. Sweeney, "Achieving K-Anonymity Privacy Protection Using Generalization and Suppression," http://en.scientificcommons.org/43196131, 2002.