Data Leakage Detection

Ahirrao P. P., Rai S. S., Pathania B. R.

Abstract: ***A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). If the data distributed to third parties is found in a public/private domain then finding the guilty party is a nontrivial task to distributor. Traditionally, this leakage of data is handled by water marking technique which requires modification of data. To overcome the disadvantages of using watermark [2], data allocation strategies are used to improve the probability of identifying guilty third parties. In this project, we implement and analyze a guilt model that detects the agents using allocation strategies without modifying the original data. The guilty agent is one who leaks a portion of distributed data. The idea is to distribute the data intelligently to agents based on sample data request and explicit data request in order to improve the chance of detecting the guilty agents. The algorithms implemented using fake objects will improve the distributor chance of detecting guilty agents. It is observed that by minimizing the sum objective the chance of detecting guilty agents will increase. We also developed a framework for generating fake objects.***

Keywords: ***sensitive data, fake objects, data allocation strategies, data leakage, data privacy, fake record.***

I. INTRODUCTION

In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties. For example, a hospital may give patient records to researchers who will devise new treatments. Similarly, a company may have partnerships with other companies that require sharing customer data. We call owner of the data, the distributor and the supposedly trusted third parties the agents. The goal of project is to detect when the distributor's sensitive data has been leaked by agents, and show the probability for identifying the agent that leaked the data. Perturbation is a very useful technique where the data are modified and made "less sensitive" before being handed to agents. For example, one can add random noise to certain attributes, or one can replace exact values by ranges. However, in some cases, it is important not to alter the original distributor's data. For example, if an out source is doing our payroll, he must have the exact salary and customer bank account numbers. If medical researcher treating patients (as opposed to simply computing statistics), they may need accurate data for the patients. Traditionally, leakage detection is handled by watermarking, for example: 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. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious. We study unobtrusive techniques for detecting leakage of a set of objects or records [1].In existing system data leakage detection is handled by watermarking, e.g., a unique code is embedded in each distributed copy. If that copy is later discovered in the hands of an illegal party, the leaker can be identified. Watermarks can be very useful in some cases, but again, involve some modification of the original data. In addition, watermarks can sometimes be cracked if the data recipient is malicious. In this paper, we study unobtrusive techniques for detecting leakage of a set of objects or records [8].

Revised Manuscript Received on 30 March 2014.
* Correspondence Author

Miss. Ahirrao P. P.*, Computer Department, SCSCOE, Rahuri. Ahmednagar, India.
Mr. Rai S. S., Computer Department, SCSCOE, Rahuri. Ahmednagar, India.
Mr. Pathania B. R., Computer Department, SCSCOE, Rahuri. Ahmednagar, India.

Objective

- The objective of the system is to detect when the distributor's sensitive data has been leaked by hackers, and if possible to identify the agent that leaked the data.
- A data infringe is the inadvertent release of secure information to an un-trusted environment.
- The goal is to estimate the likelihood that the leaked data came from the agents as opposed to other sources.
- Not only to we want to estimate the likelihood the agents leaked data, but we would also like to find out if one of them in particular was more likely to be the leaker with large number of overlapping.
- The data allocation strategies help the distributor "cleverly" give data to agents.
- Fake objects are added to identify the guilty part, to address this problem four instances are specified.
- Depending on which the data request is provided.
- Depending upon the type of data request, the fake objects are allowed.

II. EXISTING SYSTEM

In existing system data leakage detection is handled by watermarking, e.g., 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. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious. E.g. A hospital may give patient records to researchers who will devise new treatments. Similarly, a company may have partnerships with other companies that require sharing customer data. Many times agent get to know that the data will be watermark that time the data will be erase by the agent that time distributor never knows that who is the leaker.Another enterprise may out source its data processing, so data must be given to various other companies. We call the owner of the data the distributor and the supposedly trusted third parties the agents. In many cases distributor must indeed work with agents that may not be trusted, and distributor may not be sure that a leaked object came from an agent or from some other source, since sure data cannot admit watermarks.

In existing system there is few problem like fixed agents and existing system work comparable with agents whose request known in advance. Also with adding fake object original sensitive data cannot be alter and absences of agent guilt models that capture leakage scenarios and appropriate model for cases where agents can collude and identify fake tuples. Lastly system is not online capture of leak scenario also in existing system more focus on data allocation problem. Restaurant Finder Application: The prime objective of this application was to create a fully edged Android application this could locate a list of restaurants based on the location and type of the cuisine entered by the user. The user not only finds all the restaurants in the city, but also he can make a choice of the best restaurant based on the rating and cuisine he chooses to have. The user can also map the location of the restaurant on Google Maps rendered to the user on the phone and find the path from his current location or from any other location to the restaurant; the user has the facility to make a call directly to the restaurant and can also obtain the detailed review of the restaurant he chooses.[5]

Drawbacks of Existing System

Watermarks can sometimes be destroyed by the agent if the data recipient is malicious. i.e. Agent can easily remove it using various software which can easily remove watermarking from the data. There is no way to intimate the distributor when the data is leaked. In existing system there is few problem like fixed agents and existing system work comparable with agents whose request known in advance.

Future Work in Existing System

Future work includes the investigation of agent guilt models that capture leakage scenarios. For example, what is the appropriate model for cases where agents can collude and identify fake tuples? Another open problem is the extension of our allocation strategies so that they can handle agent requests in an online fashion.

III. PROPOSED SYSTEM

Our goal is to detect the guilty agent, when the distributors sensitive data has been leaked by agents, and if possible to identify the agent that leaked the data. Perturbation is a very useful technique where the data is modified and made less sensitive before being handed to agents. We propose to develop unobtrusive techniques for detecting leakage of a set of objects or records.

In this we propose to 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 acts as a type of watermark for the entire set, without modifying any individual members. 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.

Following are the points to be implemented in the proposed system to be implemented:

- Implementation of module in full security that no one can crack like existing system
- Implementation of algorithm over the existing watermarking system.
- Implementation of Fake object module for catching guilty agent.

IV. SYSTEM ARCHITECTURE

In every enterprise, data leakage is very serious problem faced by it. An owner of enterprise has given sensitive data to its employee but in most of the situation employee leak the data. That leak data found in unauthorized place such as on the web of comparator enterprise or on laptop of employee of comparator enterprise or the owner of comparators laptop. It is either observed or sometimes not observed by owner. Leak data may be source code or design specifications, price lists, intellectual property and copy rights data, trade secrets, forecasts and budgets. In this case the data leaked out it leaves the company goes in unprotected the influence of the corporation. This uncontrolled data leakage puts business in a backward position. Suppose employee given data outside the enterprise for that we devolved second model for assessing the "guilt" of agents. Guilt model are used to improve the probability of identifying guilty third parties.

The modules of the current system are listed as follows:

1. Data Allocation Module:
   - 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.
2. Fake Object Module:
   - Fake objects are generated by distributor in order to increase the chance of detecting the agents that leak the data.
   - Our use of fake objects is inspired by the use of "trace" records in mailing lists.
3. Optimization Module:
   - The Optimization Module is the distributor's data allocation to agents has one constraint and one objective.
4. Data Distributor Module:
   - A data distributor distributes his data as per the agent request.

A. System Architecture Block Diagram

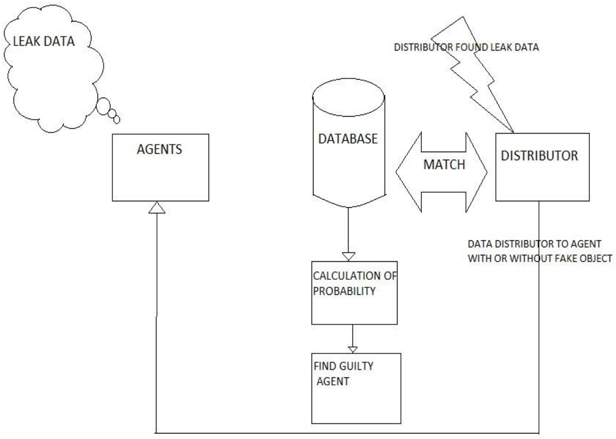

LEAK DATA
DISTRIBUTOR FOUND LEAK DATA
AGENTS
DATABASE
MATCH
DISTRIBUTOR
DATA DISTRIBUTOR TO AGENT WITH OR WITHOUT FAKE OBJECT
CALCULATION OF PROBABILITY
FIND GUILTY AGENT

Fig. 1 System Architecture Block Diagram

Problem Setup and Notation:

A distributor owns a set $T=\{t_{1,\ldots,}t_{m}\}$of valuable data objects.

The distributor wants to share some of the objects with a set of agents $U_1,U_2,\ldots U_n$, but does not wish the objects be leaked to other third parties. The objects in T could be of any type and size, e.g., they could be tuples in a relation, or relations in a database. An agent Ui receives a subset of objects, determined either by a sample request or an explicit request:

1. Sample request
2. Explicit request

V. DATA ALLOCATION STRATEGIES

In this section the allocation strategies [1] solve exactly or approximately the scalar versions. In this project, we implement and analyze a guilt model that detects the agents using allocation strategies without modifying the original data. The guilty agent is one who leaks a portion of distributed data [12]. The idea is to distribute the data intelligently to agents. We describe allocation strategies that solve exactly or approximately the scalar versions of approximation equation. We resort to approximate solutions in cases where it is inefficient to solve accurately the optimization problem.

- **Flow Chart:**

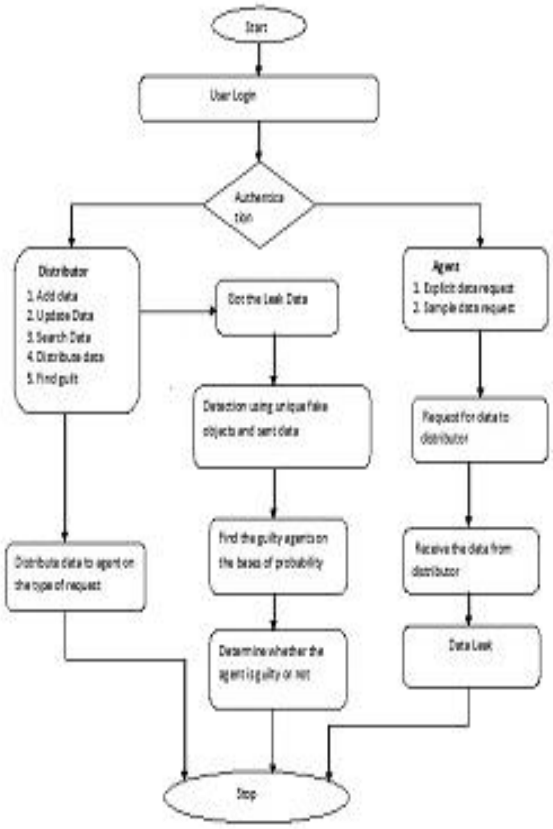

Start
User Login
Authentica tion
Distributor
1. Add data
2. Update Data
3. Search Data
4. Distribute data
5. Find guilt
Got the Leak Data
Agent
1. Explicit data request
2. Sample data request
Detection using unique fake objects and sent data
Request for data to distributor
Distribute data to agent on the type of request
Find the guilty agents on the basis of probability
Receive the data from distributor
Determine whether the agent is guilty or not
Data Leak
Stop

Fig. Flow chart for data leakage detection

A. *Explicit Data Request*

In case of explicit data request with fake not allowed, the distributor is not allowed to add fake objects to the distributed data. So Data allocation is fully defined by the agents data request. In case of explicit data request with fake allowed, the distributor cannot remove or alter the requests from the agent.

B. *Sample Data Request*

With sample data requests, each agent may receive any T from a subset out of different ones. Hence, there are different 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.

VI. *SYSTEM WORKING*

- **Module**

1. Data Allocation Module

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, Admin can send the files to the authenticated user, users can edit their account details etc. Agent views the secret key details through mail. In order to increase the chances of detecting agents that leak data.

2. Fake Object Module

The distributor creates and adds fake objects to the data that he distributes to agents. Fake objects are objects generated by the distributor in order to increase the chances of detecting agents that leak data. The distributor may be able to add 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. In case we give the wrong secret key to download the file, the duplicate file is opened, and that fake details also send the mail. Ex: The fake object details will display.

3. Optimization Module

The Optimization Module is the distributor's data allocation to agents has one constraint and one objective. The agent's constraint is to satisfy distributor's 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. User can able to lock and unlock the files for secure.

4. Data Distributor

A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). Some of the data is leaked and found in an unauthorized place (e.g., on the web or somebody's laptop). The distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. Admin can able to view which file is leaking and fake user's details also.

VI. ALGORITHMIC STUDY

1. Evaluation of explicit data request:

In the first place, the goal of these experiments was to see whether fake objects in the distributed data sets yield significant improvement in our chances of detecting a guilty agent. In the second place, we wanted to evaluate our e-optimal algorithm relative to a random allocation.

Note:

A distributor owns a set T=[t1...... tn] of valuable data objects. The objects in T could be of any type and size, e.g., they could be tuples in a relation, or relations in a database.

The distributor wants to share some of the objects with a set of agents U1;U2; ... ;Un, but does not wish the objects be leaked to other third parties

A distributor creates a set F= [F1......Fn] of fake data objects.

Algorithm: Allocation for Explicit Data Requests (EF)
Input: T1, ... , Tn, cond1, ... ,condn, b1,...,bn, B
Output: T1; ...; Tn, F1; ... ; Fn

Steps

1. Calculate total fake records as sum of fake Records allowed.
2. While total fake objects = 0.
3. Select agent that will yield the greatest improvement in the sum objective.
4. Create fake record.
5. Add this fake record to the agent and also to fake record set.
6. Decrement fake record from total fake record set.

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

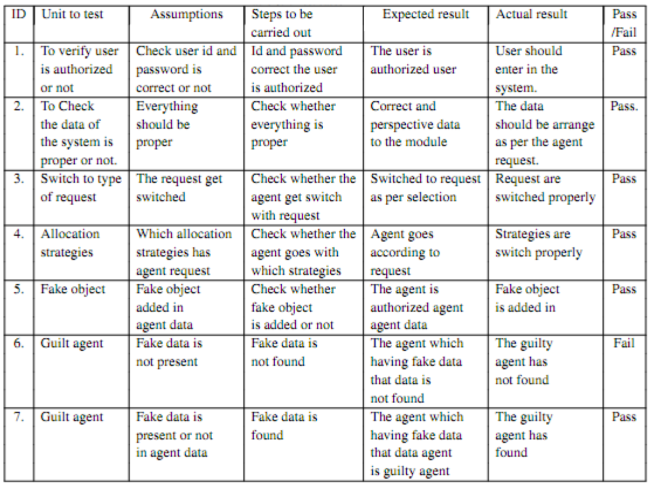
2. Evaluation of Sample data request:

With sample data requests agents are not interested in particular objects. Hence, object sharing is not explicitly defined by their requests. The distributor is ”forced” to allocate certain objects to multiple agents only if the number of requested objects exceeds the number of objects in set T. The more data objects the agents request in total, the more recipients on average an object has; and the more objects are shared among different agents, the more difficult it is to detect a guilty agent.

Algorithm: Allocation for Sample Data Requests (SF)
Input: m1; ... ; mn,[T]. Assuming mi¡= [T]
Output: T1; ... ;Tn
Steps

1. The distributor gives the data to agents such that he can easily detect the guilty agent in case of leakage of data.
2. To improve the chances of detecting guilty agent, he injects fake objects into the distributed data set.
3. These fake objects are created in such a manner that, agent cannot distinguish it from original objects.
4. One can maintain the separate data set of fake objects orcan create it on demand.

Test Cases:



ID	Unit to test	Assumptions	Steps to be carried out	Expected result	Actual result	Pass /Fail
1.	To verify user is authorized or not	Check user id and password is correct or not	Id and password correct the user is authorized	The user is authorized user	User should enter in the system.	Pass
2.	To Check the data of the system is proper or not.	Everything should be proper	Check whether everything is proper	Correct and perspective data to the module	The data should be arrange as per the agent request.	Pass.
3.	Switch to type of request	The request get switched	Check whether the agent get switch with request	Switched to request as per selection	Request are switched properly	Pass
4.	Allocation strategies	Which allocation strategies has agent request	Check whether the agent goes with which strategies	Agent goes according to request	Strategies are switch properly	Pass
5.	Fake object	Fake object added in agent data	Check whether fake object is added or not	The agent is authorized agent agent data	Fake object is added in	Pass
6.	Guilt agent	Fake data is not present	Fake data is not found	The agent which having fake data that data is not found	The guilty agent has not found	Fail
7.	Guilt agent	Fake data is present or not in agent data	Fake data is found	The agent which having fake data that data agent is guilty agent	The guilty agent has found	Pass

The above all are test cases which we used in our paper.

VII. EXPECTED RESULT

We implemented the presented allocation algorithms in Python and we conducted experiments with simulated data leakage problems to evaluate their performance.

A. Experiments need to conduct analysis result:

In our scenarios we have taken a set of 500 objects and requests from every agent are accepted. There is no limit on number of agents, as we are considering here their trust values [3]. The flow of our system is given as below:

1. Agents Request: Either Explicit or Implicit.
2. Leaked dataset given as an input to the system.
3. The list of all agents having common tuples as that of leaked tuples is found and the corresponding guilt probabilities are calculated.
4. It shows that as the overlap with the leaked dataset minimizes the chances of finding guilty agent increases.

In our paper we trying to give the exact output to the user that not occur any error. In this paper the distributer is finding the guilty agent and tries to minimize the data leakage detection.

VIII. ANALYSIS

We present the metrics we use for the algorithm evaluation. We present the evaluation for sample requests and explicit data requests, respectively. We presented algorithms to optimize the problem of that is an approximation to the original optimization problem of. We evaluate the presented algorithms with respect to the original problem. In this way, we measure not only the algorithm performance, but also we implicitly evaluate how effective the approximation is.

The data distribution strategies improve the distributors chances of identifying a leaker. It has been shown that distributing objects judiciously can make a significant difference in identifying guilty agents, especially in cases where there is large overlap in the data that agents must receive.

In some cases “original but fake” data records are injected to improve the chances of detecting leakage and identifying the guilty party. Our future work includes extension of this work considering allocation strategies so that they can handle agent requests uniquely in an online fashion and fake data using encryption techniques. However, in many cases we must indeed work with agents that may not be 100 percent 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 objects can be “guessed” by other means. In the first place, the goal of these experiments was to see whether fake objects in the distributed data sets yield significant improvement in our chances of detecting a guilty agent.

In the second place, we wanted to evaluate our e-optimal algorithm relative to a random allocation. We focus on scenarios with a few objects that are shared among multiple agents. These are the most interesting scenarios, since object sharing makes it difficult to distinguish a guilty from non-guilty agents. Scenarios with more objects to distribute or scenarios with objects shared among fewer agents are obviously easier to handle. As far as scenarios with many objects to distribute and many overlapping agent requests are concerned, they are similar to the scenarios we study, since we can map them to the distribution of many small subsets.

IX. ADVANTAGE AND APPLICATION

1. ADVANTAGES

- Using the technique of Perturbation data is made less sensitive for the agents to handle.
- Realistic but fake objects are injected to the distributed data set to identify the guilt agent.
- If two agent have same probability then the FIFO order is maintained to show the guilty agent.
- Possibility of full service with maintenance and SLA in overall service.
- Easier access to new software versions.

2. APPLICATION

- It provides data secrecy and identify the guilt agent in case of data leakage.
- We present is related to the data provenance problem.
- Our work is mostly relevant to watermarking that is used as a means of establishing original ownership of distributed objects.
- Our approach and watermarking are similar in the sense of providing agents with some kind of receiver identifying information.
- The goal of these experiments was to see whether fake objects in the distributed data sets yield significant improvement in our chances of detecting a guilty agent.

X. CONCLUSION

From this project we conclude that the data leakage detection system model is very useful as compare to the existing watermarking model. We can provide security to our data during its distribution or transmission and even we can detect if that gets leaked. Thus, using this model security as well as tracking system is developed. Watermarking can just provide security using various algorithms through encryption, whereas this model provides security plus detection technique. This model is very helpful in various industries, where data is distribute through any public or private channel and shred with third party. Now, industry and various offices can rely on this security and detection model. Data leakage is a silent type of threat. Your employee as an insider can intentionally or accidentally leak sensitive information. This sensitive information can be electronically distributed via e-mail, Web sites, FTP, instant messaging, spreadsheets, databases, and any other electronic means available all without your knowledge. To assess the risk of distributing data two things are important, where first one is data allocation strategy that helps to distribute the tuples among customers with minimum overlap and second one is calculating guilt probability which is based on overlapping of his data set with the leaked data set.

XI. ACKNOWLEDGMENT

Students work is incomplete until they thank the All mighty & his teachers. I sincerely believe in this and would like to thank Prof. Jadhav Sir, Head of the Department, Computer Science & Engineering, SCSCOE, Pune for hem encouragement and motivation to write this paper. Also I am grateful to Prof. Dighe sir, Pune for guiding me in writing this paper. And all of the most is my project partner without them we can no complete our project and paper

REFERENCES

1. Panagiotis Papadimitriou, Student Member, IEEE, and Hector Garcia-Molina, “DATA LEAKAGE DETECTION ”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 1, JANUARY 2011.
2. Sandip A. Kale, Prof. S.V.Kulkarni,Department Of CSE, MIT College of Engg, Aurangabad,“Data Leakage Detection: A Survey ”,Vol. 1, NO. 6, July-Aug 2012.
3. Prerna Jawdand, Prof. Girish Agarwal, Prof. Pragati Patil Student (Mtech-CSE)AGPCE, Professor (Mtech-CSE)AGPCE, HOD (Mtech-CSE)AGPCE, “DATA LEAKAGE DETECTION ”, International Journal of Engineering Research and Technology (IJERT) ISSN: 2278-0181 Vol. 2 NO. 1, January- 2013.
4. Mr.V.Malsoru, Naresh Bollam “Review On Data Leakage Detection ”, International Journal Of Engineering Reserach And Applications(IJERA) VOL.1,NO. 3,May- 2013.
5. Panagiotis Papadimitriou, Hector Garcia-Molina “Data Allocation Strategies ”,International Journal On Trends And Technology,Vol. 3,NO. 4, April 2012,.
6. Archana Vaidya, Prakash Lahange, Kiran More, Shefali Kachroo and Nivedita Pandey “Data Leakage Detection ” International Journal of Advances in Engineering and Technology,VOL. 2231, NO. 1963, March 2012.
7. Amir Harel, Asaf Shabtai, LiorRokach, and Yuval Elovici “AMisuseabilityWeightMeasure”, IEEE Transactions ON Dependable And Secure Computing, Vol.9, NO. 3, MAY/JUNE 2012.
8. Rohit Pol, Vishwajeet Thakur, Ruturaj Bhise, Prof. Akash Kate “Data Leakage Detection ”, International Journal of Engineering Research and Applications (IJERA),VOL. 2, NO. 3, May-Jun 2012.
9. Rudragouda G Patil, “Development of Data leakage Detection Using Data Allocation Strategies ”, International Journal of Computer Applications in Engineering Sciences, VOL. I, NO. II, JUNE 2011.
10. Unnati Kavali,Tejal Abhang, Mr.Vaibhav Narawade, “Data Allocation Strategies In Data Leakage Detection”, International Journal Of Engineering Reserach And Applications(IJERA), VOL 2,NO. 2,May-2011.
11. Polisetty Sevani Kumari,Kavidi Venkata Mutyalu,Development Of Data Leakage Detection Using Data Allocation, VOL. 2,NO. 2, Jun 2011.