DATA LEAKAGE DETECTION

NIKHIL CHAWARE [1], PRACHI BAPAT [2], RITUJA KAD [3], ARCHANA JADHAV [4], PROF.S.M.SANGVE [5]

[1,2,3,4]Student Computer Department,ZES's DCOER,Pune
[5]Assistant Professor and HOD Computer Department,ZES's DCOER,Pune

nikhil4u28@gmail.com, shriyaabapat@gmail.com, ritukad19@gmail.com,jadhavarchana11@gmail.com

ABSTRACT *: In the business, sometimes sensitive data must be handed over to trusted third parties. So some companies distribute their data to trusted third parties. These companies (data distributer) found their some of the data in unauthorized place (e.g., on the web or somebody's laptop). The distributor understands that the leaked data came from one or more agents. Our goal is to detect which agent leaks that data and provide the security to that data. When the distributor's sensitive data have been leaked by agents, and to identify the agent that leaked the data. We propose data allocation strategies (across the agents) that improve the probability of identifying leakages. These methods do not 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. The Main Aim of the system can be given as follows:-*
Identify data leakages from distributed data using some data allocation strategies and find out the fake agent who leak that data. Improve probability of finding out fake agent and provide the security to that data.

***Keywords*— DLD(Data Leakage Detection)**

I. INTRODUCTION

Providing security to the data is important because sometimes sensitive data must be given to trusted third parties. For example, a company may have partnerships with other companies that require sharing customer data. Another enterprise may outsource its data processing, so data must be given to various other companies. It may possible to leak their data through the third parties [1]. We call the owner of the data the distributor and the trusted third parties the agents.
Traditionally, 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. In this, we study unobtrusive techniques for detecting leakage of a set of objects or records.
Specifically 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 [2]. 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. 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.

II. LITERATURE SURVEY

Leakage detection system has proposed by Panagiotis Papadimitriou; Hector Garcia Molina which can enable us to detect the guilty leaker without changing the integrity of the original data.
An enterprise data leak is a scary proposition. Security practitioners have always had to deal with data leakage issues that arise from email and other Internet channels. But now with the use of mobile technology, it's easier for data loss to occur, whether accidentally or maliciously.
The guilt detection approach we present is related to the data provenance problem [3],[5]: tracing the lineage of S objects implies essentially the detection of the guilty agents. And assume some prior knowledge on the way a data view is created out of data sources. Our problem formulation with objects and sets is more general As far as the data allocation strategies are concerned; our work is mostly relevant to watermarking (stegnography) that is used as a means of establishing original ownership of distributed objects. Finally, there are also lots of other works on mechanisms that allow only authorized users to access sensitive data through access control policies [4]. Such approaches prevent in some sense data leakage by sharing information only with trusted parties. However, these policies are restrictive and may make it impossible to satisfy agent's requests.

Existing System:
In existing system, we consider applications where the original sensitive data cannot be made less sensitive . However, in some cases it is important not to alter the original distributor's data. Traditionally, leakage detection is handled by giving a unique code and it is embedded within distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified data. 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.

III. PROPOSED METHOD

In proposed System, we can remove disadvantages of watermarking by adding fake objects thus increasing efficiency of the system, 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. If the distributor sees "enough evidence" that an agent leaked data, he may stop doing business with him, or may initiate legal proceedings. In this project 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, adding "fake" objects to the distributed set, do not correspond to real entities but appear. 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. Our approach and watermarking are similar in the sense of providing agents with some kind of receiver identifying information. However, by its very nature, a watermark modifies the item being watermarked. If the object to be watermarked cannot be modified, then a watermark cannot be inserted. In such cases, methods that attach watermarks to the distributed data are not applicable.

Algorithms
Allocation for Explicit Data Requests: In this request the agent will send the request with appropriate condition. Agent gives the input as request with input as well as the condition for therequest after processing the data after processing on the data the gives the data to agent by adding fake object with an encrypted format. Allocation for Sample Data Requests: In this request agent request does not have condition. The agent sends the request without condition as agent sends the request without condition as per his query he will get the data. 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. 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.

IV. RESULT

We implemented the presented allocation algorithms, and we conducted experiments with simulated data leakage problems to evaluate their performance. Data leakage poses a serious issue for companies as the number of incidents and the cost to those experiencing them continue to increase. Data leakage is enhanced by the fact that transmitted data (both inbound and outbound), including emails, instant messaging, website forms, and file transfers among others, are largely unregulated and unmonitored on their way to their destinations.
Our project comes into NP complete because in particular time it will give the result. For the decision problem, so that it will give the solution for the problem within polynomial time

CONCLUSION

Sensitive Data can be leaked by the agents unknowingly or maliciously and even if we had to hand over sensitive data , In a perfect world we could use the concept of Stegnography so that we could add Fake objects.
To detect the agent who have leaked the data. The algorithm we have presented implement the variety of data distribution strategies that can improve the distributors chances of identifying a leakar.

REFERENCE

[1] Panagiotis Papadimitriou, Hector Garcia-Molina , IEEE Paper "Data Leakage Detection",2011.
[2] Panagiotis Papadimitriou, Hector Garcia-Molina , IEEE Paper "Data Leakage Detection",2010.
[3] P. Buneman, S. Khanna, and W.C. Tan, "Why and Where: A Characterization of Data Provenance," Proc. Eighth Int'l Conf. Database Theory (ICDT '01), J.V. den Bussche and V. Vianu, eds., pp. 316-330, Jan. 2001.
[4] P. Bonatti, S.D.C. di Vimercati, and P. Samarati, "An Algebra for Composing Access Control Policies," ACM Trans. Information and System Security, vol. 5, no. 1, pp. 1-35, 2002.
[5] P. Buneman and W.-C. Tan, "Provenance in Databases," Proc. ACM SIGMOD, pp. 1171-1173, 2007.