International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 1, January- 2013

# Implementation Of Data Leakage Detection Using Agent Guilt Model

**Prerna Jawdand, Prof. Girish Agarwal, Prof. Pragati Patil**

**Student (Mtech-CSE)AGPCE, Professor (Mtech-CSE)AGPCE, HOD (Mtech-CSE)AGPCE**

***ABSTRACT-***

A distributor has given sensitive data to some supposedly faithful agents. Sometimes data is leaked and found in unauthorized place e.g., on the web or on somebody's laptop. Consider a hospital may give patient records to researchers who will devise new treatments. Same way, a company nowadays has partnerships with other companies that require sharing customer data. Some different enterprise may outsource its data processing, so data might be given to various other companies. The owner of the data is called as *distributors* and the trusted third parties are called as *agents.* Data leakage happens when confidential business information such as customer or patient data, company secrets, budget information etc. is leaked out. When this information is leaked out, then the companies are at serious risk. Most probably data are being leaked from agent's side. So, company have to very careful while giving data to various agents. The Goal of my project is to test carefully "how the distributer can allocate the private data to the Agents so that the leakage of data would be minimized to a Greater space by finding an guilty agent".

***Index Terms*- data, distributor, leakage, sensitive data**

## INTRODUCTION

Sometimes sensitive data must have be handed over to supposedly trusted third parties. Let us consider, a hospital may give patient records to researchers who will devise new treatments. Same way, a company may have partnerships with other companies that require sharing customer data. Different other enterprise may outsource its data commanding, so data have to be given to various other companies. We can say the owner of the data the distributor and the trusted third parties the agents. Main Goal is to detect when the distributor's sensitive data has been leaked by agents, and possibly to identify the agent that leaked the data. We consider several uses where the original sensitive data cannot be disturbed. Disturbance is a very useful technique where the data is modified and made "less sensitive" before being handed to other agents. For example, one can add random noise to certain attributes, or one can exchange exact values by ranges. But, in some cases it is important not to alter the original distributor's data. Consider if an outsourcer is doing our payroll, he must have the exact salary amount and customer bank account numbers. Even if medical researchers will be treating patients they may need accurate data for the patients.

Traditionally, leakage detection is handled by watermarking, where we can say a unique code is embedded in each distributed copy. A design impressed in some paper during manufacture is nothing but watermarks. If this copy is later discovered in the hands of an unauthorized party, the leaker is supposed

to be found out. Watermarks can be very useful in some cases, involve some changes of the original data. Watermarks can sometimes be ruined if the data recipient is spiteful. I will develop a model for assessing the "guilt" of agents. I also present algorithms for distributing objects to agents, in a way that improves our chances of detecting a leaker. Finally, also consider the option of adding "fake" objects to the given set of data. Such things do not respond to real entities but appear lifelike to the agents. The fake objects acts as a type of impressed design for the entire set, without changing any individual members. If it shows an agent was given one or more fake objects that were leaked, in such a case the distributor can be stronger that agent was guilty. I will evaluate the strategies in different data leakage outline, and check whether they indeed help us to identify a leaker.

## LITERATURE REVIEW

### 1) Identifying Guilty Agents

ArchanaVaidya[2] presented a model which is relatively simple, but believed that it captures the essential trade-offs. The algorithms that have presented implement a variety of data distribution strategies that can improve the distributor's chances of identifying a leaker. Authors have 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.

### 2) Guessing an Agent

Panagiotis Papadimitriou[1] presents that 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 chance that things can be "conjectured" by other means.

### 3) A Misuse ability Weight Measure

Amir Harel [3] introduced a new concept of misuse ability weight and discussed the importance of measuring the sensitivity level of the data that an insider is exposed to. Here defined four dimensions that a wrong usability weight measure must consider. To the best of the knowledge and based on the literature survey done, there is no previously proposed method for estimating the potential harm that might be caused by leaked or wrong used data while taking into account important dimensions of the nature of the exposed data. Consequently, a new wrong usability measure, the M-score, was proposed. Extended the M-score basic definition to consider prior knowledge the user might have and presented four applications using the extended definition. Finally, explored different approaches for efficiently acquiring the knowledge required for computing the M-score, and showed that the M-score is both practicable and can fulfill its main goals.

### 4) Development of Data leakage Detection Using Data Assignment Methods

In doing a business there would be no need to hand over sensitive data to agents that may unknowingly or harmfully leak it. And even if we had to hand over sensitive data, in a perfect world we could impressed a design of each object so that we could trace

its origins with absolute certainty. But 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, this paper have shown it is possible to assess the likelihood that an agent is liable for a leak, based on the overlap of his data with the leaked data and the data of other agents, and based on the chance that objects can be "guessed" by other means.

**5)Assessing the guilt of agents**

In proposed system, after giving a set of objects to agents, the distributor find out some of those same objects in a futile 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 freely gathered by other means. If the distributor sees enough evidence that an agent leaked data, he may be stop doing business with him, or may start legal action. In this project the author develop a model for assessing the guilt of agents and also present algorithms for distributing objects to agents, in a way that improves our chances of detecting a leaker. Finally, also consider the option of adding fake objects to the divided set. Such things do not correspond to actual entities but seem. If it shows an agent was given one or more fake objects that were leaked, then the dealer can be more confident that agent was guilty.

**EXISTING WORK**

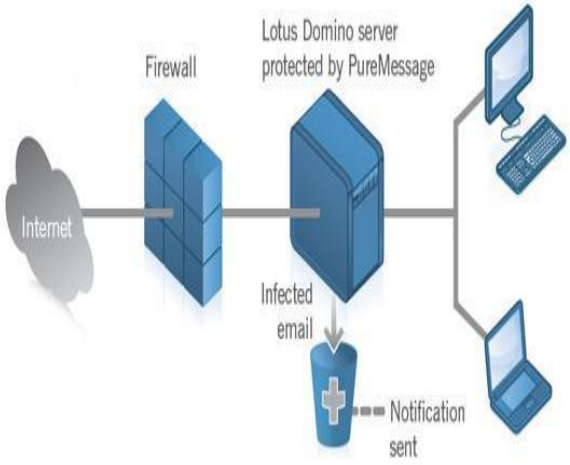
Previously, leakage detection is handled by impressing a design. Here a unique code is embedded in each distributed copy. If this copy is discovered in the hands of an unauthorized party, the leaker may be identified. Watermarks can be very useful in some cases, involve some modification of the original data. Watermarks can sometimes be ruined if the data recipient is spiteful. *Say* A hospital may give patient records to researchers who will devise new treatments. Same way, a company may have partnerships with other companies that require sharing customer data. Other enterprise may outsource its data processing; hence data must be given to various other companies. We can call the owner of the data the distributor and the supposedly trusted third parties the agents.

**OVERVIEW OF PROPOSED WORK**

My goal is to detect when the distributor's sensitive data has been leaked by agents, and if possible to identify the agent who leaked the data. Disturbance is a very useful technique. Here the data is altered and made "less effective" before being handed to agents. We develop *modest* techniques for detecting leakage of a set of objects or records.

We also create a model for assessing the "guilt" of agents. We present algorithms for distributing objects to agents, in such way that improves our chances of identifying a leaker. We also ponder the option of adding "defraud" objects to the distributed set. These objects do not correspond to real entities but appear realistic to the agents. The fake objects acts as a type of watermark for the complete set, without modifying any individual members. If it shows an agent was given one or more fake objects that were leaked, then the distributor will be

more confident that particular agent was guilty.




## 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*

## GUILT MODEL ANALYSIS

Our model parameters interact and to check if the interactions match our insight, We study two simple cases as Impact of Probability p and Impact of Overlap between Ri and S. In each case we have to face that has obtained all the distributor's objects, i.e., T = S.

## ALGORITHMS

### 1.Evaluation of Explicit Data Request Algorithms

The aim of these practical is to see whether fake things in the distributed data sets yield significant improvement in our chances of detecting a guilty agent. We also wanted to think out our e-optimal algorithm relative to a random allocation.

### 2. Evaluation of Sample Data Request Algorithms

With sample data requests agents are not amusing in particular things. Object sharing is not definitely defined by their petition. The distributor is "overstrained" to assign certain things to many agents only if the number of requested things exceeds the number of things in set T. The more data objects the agents request in total, the more people to receive on average an object has; and the more objects are shared among different various agents, the more difficult it is to find out a guilty agent.

## MODULES STRUCTURE

### 1. Data Allocation Module

The main attention of our project is the data distribution problem as how can the distributor "intelligently" give data to agents in order to improve the chances of detecting a guilty agent.[7]

### 2. Fake Object Module

Fake objects are objects created by the distributor in order to increase the chances of detecting agents those who are responsible to leak data. The distributor could be able to add unwanted objects to the distributed data in order to improve his effectiveness in detecting guilty agents. Our use of fake objects is influenced by the use of "trace" records in mailing lists.

### 3. Optimization Module

The Optimization Module is the distributor's data assignment to agents has one constraint and one objective. The distributor's compulsion is to satisfy agents' requests, by giving them with the number of objects they want or with all available objects that satisfy their needs. His aim is to be able to detect an agent who leaks any portion of his data.

### 4. Data Distributor

A data distributor has given important data to a set of supposedly trusted agents. This data is leaked and found in an unsantioned place (e.g., on the web or somebody's personal computers). The distributor must assess the likelihood that the leaked data came from one or more agents.

## CONCLUSION

In a perfect world there must not be any need to hand over sensitive data to agents that may unknowingly or maliciously leak it. And if we have to hand over sensitive data, in a perfect world we could impressed a design in each object so that we could trace its origins with perfect certainty. However, in many cases we must indeed work with agents that may not be fully trusted, and we may be uncertain if a leaked object came from an agent or from some other source, as certain data cannot admit watermarks.

In spite of such difficulties, earlier it is shown that it is possible to assess the likelihood that an agent is responsible for a leak, based on the overlap of agent data with the leaked data and the data of different agents, and based on the probability that objects can be "guessed" by other means. My model is quite simple, but we believe it captures the important trade-offs. Future work of this paper 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 (the presented strategies assume that there is a fixed set of agents with requests known in advance).



## REFERENCES

[1]Panagiotis Papadimitriou, Student Member, IEEE, and Hector Garcia-Molina, Member, IEEE "Data Leakage Detection" IEEE Transactions on Knowledge and Data Engineering, Vol. 23, NO. 1, JANUARY 2011

[2]Archana Vaidya, Prakash Lahange, Kiran More, Shefali Kachroo & Nivedita Pandey "Data Leakage Detection" International Journal of Advances in Engineering & Technology, March 2012 ©IJAETISSN: 2231-1963

[3]Amir Harel, Asaf Shabtai, LiorRokach, and Yuval Elovici "M-Score: A Misuseability Weight Measure" IEEE Transactions ON Dependable And Secure Computing, Vol.9, NO. 3, MAY/JUNE 2012

[4]Rudragouda G Patil, "Development of Data leakage Detection Using Data Allocation Strategies International Journal of Computer Applications in Engineering Sciences [Vol I, ISSUE II, JUNE 2011, [ISSN: 2231-4946]

[5]Rohit Pol, Vishwajeet Thakur, Ruturaj Bhise, Prof. Akash Kate "Data Leakage Detection" International Journal of Engineering Research and Applications (IJERA)Vol. 2, Issue 3, May-Jun 2012, pp. 404-410 ISSN: 2248-9622

[6] Mr.V.Malsoru, Naresh Bollam "Review On Data Leakage Detection" International Journal Of Engineering Reserach And Applications(IJERA) Vol.1,Issue 3,pp.1088-1091 ISSN: 2248-9622

[7]Unnati Kavali,Tejal Abhang, Mr.Vaibhav Narawade "Data Allocation Strategies In Data Leakage Detection" International Journal Of Engineering Reserach And Applications(IJERA) Vol 2,Issue 2,pp.1448-1452 ISSN: 2248-9622

[8]Polisetty Sevani Kumari,Kavidi Venkata Mutyalu,"Development Of Data Leakage Detection Using Data Allocation Strategies"International Journal On Trends And Technology,Vol 3,Issue 4-2012