Efficiently Identifying the Data Leakages

[1]K. Sudhakara Babu, [2]P. Sudhakar Babu
[1,2]Dept. of CSE, Lenora College of Engineering –Rampa Chodavaram

Abstract
Data leakage is defined as the accidental or unintentional distribution of private or sensitive data to an unauthorized entity. Sensitive data in companies and organizations include intellectual property (IP), financial information, patient information, personal credit-card data, and other information depending on the business and the industry. 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. Furthermore, in many cases, sensitive data are shared among various stakeholders such as employees working from outside the organization's premises (e.g., on laptops), business partners, and customers. This increases the risk that confidential information will fall into unauthorized hands. Whether caused by malicious intent or an inadvertent mistake by an insider or outsider, exposure of sensitive information can seriously hurt an organization.The Data Leakage problem can be defined as any unauthorized access of data due to an improper implementation or inadequacy of a technology, process or a policy. The "unauthorized access" described above can be the result of a malicious, intentional, inadvertent data leakage, or a bad business/technology process from an internal or external user. Traditionally, this leakage of data is handled by water marking technique which requires modification of data. If the watermarked copy is found at some unauthorized site then distributor can claim his ownership. 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
Data Leakage, Sensitive data, Fake Data, Data Request, Guilt Model.

I. Introduction

Demanding market conditions encourage many companies to outsource certain business processes (e.g. marketing, human resources) and associated activities to a third party. This model is referred as Business Process Outsourcing (BPO) and it allows companies to focus on their core competency by subcontracting other activities to specialists, resulting in reduced operational costs and increased productivity. Security and business assurance are essential for BPO. In most cases, the service providers need access to a company's intellectual property and other confidential information to carry out their services. For example a human resources BPO vendor may need access to employee databases with sensitive information (e.g. social security numbers), a patenting law firm to some research results, a marketing service vendor to the contact information for customers or a payment service provider may need access to the credit card numbers or bank account numbers of customers. The main security problem in BPO is that the service provider may not be fully trusted or may not be securely administered. Business agreements for BPO try to regulate how the data will be handled by service providers, but it is almost impossible to truly enforce or verify such policies across different administrative domains. Due to their digital nature, relational databases are easy to duplicate and in many cases a service provider may have financial incentives to redistribute commercially valuable data or may simply fail to handle it properly. Hence, we need powerful techniques that can detect and deter such dishonest.

We study unobtrusive 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. (For example, the data may be found on a web site, or may be obtained through a legal discovery process.) 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. 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.

A. Type of Data Leakage

In order to implement the appropriate protective measures, we must first understand what we are protecting. Based on publicly disclosed Data Leakage breaches, the type of data leaked is broken down as follows:

Table 1: Type of Information Leaked

Type of Information Leaked	Percentage
Confidential Information	15%
Intellectual Information	4%
Customer Data	73%
Health Records	8%

B. Internal Threats

According to data compiled from EPIC.org and PerkinsCoie.com, 52% of Data Security breaches are from internal sources compared to the remaining 48% by external hackers.

It increases the risk of confidential information falling into unauthorized hands. Whether caused by malicious intent, or an inadvertent mistake, by an insider or outsider, exposed sensitive information can seriously hurt an organization. The potential damage and adverse consequences of a data leak incident can be classified into the following two categories: direct and indirect loss. Direct loss refers to tangible damage that is easy to measure and estimate quantitatively. Indirect loss, on the other hand, is much harder to quantify and has a much broader impact in terms of

cost, place and time [Bunker, 2009]. Direct loss includes violating regulations (such as those protecting customer privacy) resulting in fine/settlement/customer compensation fees; litigation of lawsuits; loss of future sales; costs of investigation and remedial/restoration fees. Indirect loss includes reduced share-price as a result of the negative publicity; damage to company's goodwill and reputation; customer abandonment; and exposure of Intellectual Property (business plans, code, financial reports, and meeting agendas) to competitors.

C. Detecting Challenges

1. Encryption

and preventing data leaks in transit are hampered due to encryption and the high volume of electronic communications. While encryption provides means to ensure the confidentiality, authenticity and integrity of the data, it also makes it difficult to identify the data leaks occurring over encrypted channels. Encrypted emails and file transfer protocols such as SFTP imply that complementary DLP mechanisms should be employed for greater coverage of leak channels. Employing data leak prevention at the endpoint – outside the encrypted channel – has the potential to detect the leaks before the communication is encrypted.

2. Access Control

Access control provides the first line of defense in DLP. However, it does not have the proper level of granularity and may be outdated. While access control is suitable for data at rest, it is difficult to implement for data in transit and in use. In other words, once the data is retrieved from the repository, it is difficult to enforce access control. Furthermore, access control systems are not always configured with the least privilege principle in mind. For example, if an access control system grants full access to all code repositories for all programmers, it will not effectively detect data leaks where a programmer accesses a project that he/she is not involved in.

3. Semantic Gap in DLP

DLP is a multifaceted problem. The definition of a data leak is likely to vary between organizations depending on the sensitive data to be protected, the degree of interaction between the users and the available communication channels. The current state-of-the-art, which is reviewed in Section III, mainly focuses on the use of misuse detection (signatures) and post-mortem analysis (forensics). The common shortcoming of such approaches is that they lack the semantics of the events being monitored. When a data leak is defined by the communicating parties as well as the data exchanged during the communication, a simple pattern matching or access control scheme cannot infer the nature of the communication. Therefore, data leak prevention mechanisms need to keep track of who, what and where to be able to defend against complex data leak scenarios.

The classification by leakage channel is important in order to know how the incidents may be prevented in the future and can be classified as physical or logical.

Physical leakage channel means that physical media (e.g., HDD, laptops, workstations, CD/DVD, USB devices) containing sensitive information or the document itself was moved outside the organization. This more often means that the control over data was lost even before it leaved the organization

II. Existing System

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.

III. Proposed System

Our goal is to detect, when the distributor's 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.

To find the solution on the existing problem we develop two models. First, when any employee of enterprise access sensitive data without the consent of owner in that case, we developed data watcher model to identifying data leaker in this point suppose data leaker will identify then no need to calculating the probability of agents that method gives near about 90 % of result. But 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..

For implementing this system we used SSBT'S COET, A, B college database. In this system we consider data owner is college management called distributor and other employee is called agents. For that we take two condition sample or explicit condition because agents want data in sample or condition. In this approach, the model for assessing the ―guilt of agents is developed. The option of adding ―fake objects to the distributed set is considered. Such objects do not correspond to real entities but appear practical 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.

Proposed System worked on two processes:

A. Data Distribution Process

In that considered two exiting techniques for data allocating to the agents. There are four instances of this problem they address, depending on the type of data requests made by agents (E for Explicit and S for Sample requests) and whether ―fake data are allowed (F for the use of fake data, and F for the case where fake data are not allowed). Fake data are data generated by the distributor that are not in set T. The data are designed to look like real data, and are distributed to agents together with the T data, in order to increase the chances of detecting agents that leak data [1].

B. Probability Finding Process

While distributing the data to any agents some kind of receiver's information can be added to find out the guilty agent it is more concentrated on finding the probability of an agent to be found as guilty. Data object is to be important aspect of our work; it is

consider agents parameter and overlapping between pair of agents of this data object which we are forwarding to other agent. The parameter would then be checked once a data object is received from a malicious target for that used a special process for data object is received from any target the probability is calculated the data object came from which source or we can guess that which agent has leaked the data. Guilty Agent Model would be used to find the agent to be guilty with numerous conditions. Also we have considered if the object cannot be guessed or if its probability can't be find out then the agent can't be considered to be guilty.

C. Algorithm for Finding Guilt Agent

1. Distributor select agent to send data. Distributor selects the agents to send the data according to agent request.
2. Distributor creates fake data and allocates it to the agent.
3. 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.
4. Check number of agents, who have already received data. Distributor checks the number of agents, who have already received data.
5. Check for remaining agents. Distributor chooses the remaining agents to send the data. Distributor can increase the number of possible allocations by adding fake data.
6. 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.

D. Implementation

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.

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 [1,7].

In the figure1 Distributor has been given the data to agents according to the request by agents. In this system database maintained by Distributor according to the request by agents data passing to agents with fake or without fake object. When agent doing the business with target without the consent of distributor and leak data. Distributor discovers some of those same objects in an unauthorized place. (For example, the data may be found on a web site, or may be obtained through a legal discovery process or someone's laptop). Then distributor match leak data with his data. Distributor also check overlapping of data among agents and then he calculating probability of agents varies for {0, 1}

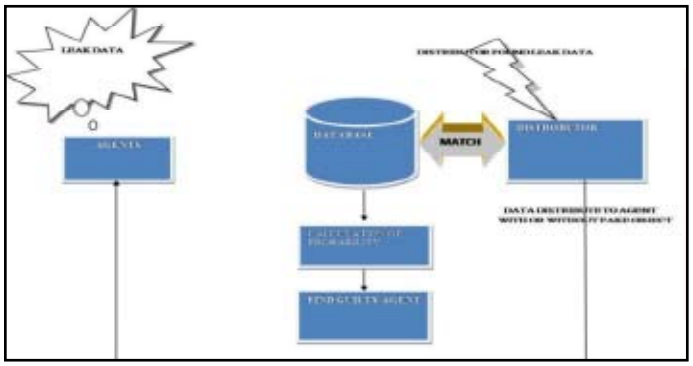

Fig. 1: Architecture of Data Leakage Detection

E. Result

The effectiveness of a system is most commonly described with its "Record wise leak report" and Probability of agent guilty.

Record wise leak report= $\frac{|T \cap S|}{|S|}$

This formula calculates records wise leak data without considering agents overlapping.

According to that the distributor knows which agents consider for calculating guilt probability.

Probability of agent guilty = $\frac{|Ri \cap S|}{|S|}$

This formula calculates records wise leak data with considering agents overlapping mean that particular record share by how many agents. According to that the distributor knows which agents consider for calculating guilt probability.

agentGuiltyProbability

agentName	ProbabiltyOfguilty	Add New Field
Agent1	0.90496	
Agent2	0.266666666666667	
Agent3	0.784	
Agent5	0.56	
*	0	

FIGURE 2: AGENT GUILT PROBABILITIES

As shown In fig. 2, given guilt probability of agents like agent1 having 0.9 probability means he having more probable to leak the owner data then agent3 having 0.7 then agent 5 having 0,5 likewise owner can find out leak scenario. As shown in Figure III First owner set estimate probability from 0.1 to 0.9 according to owner can find guilt agents.

As shown in fig. 4, according to the record distribution to agents plot the graph in that considered probability value from {0 to 1} and estimate value according to assumption 2. Generally this thing will be set by owner of enterprise according to the liability of agent owner set estimate probability for 0.1 to 0.9.

IV. Conclusion

In the enterprise owner hand over its sensitive data to employee but before that owner must be add water mark to each and every sensitive data. Also check the history of employee means that particular employee is liable or not to handle that data or not. Then hand over data to employee. Suppose that employee leak data accidently for this case owner considering the estimate probability

of employee. If employee leak data deliberately then owner think about that particular employee for not involved in shared data or confidential work or talk.

AgentProbabilityDistribution

agentName	valueOfP	Prob
Agent1	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1	0.064,0.006,0.001,0,0,0,0,0,0,0
Agent2	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1	0.7,0.513,0.394,0.315,0.262,0.227,0.205,0.191,0.1
Agent3	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1	0.166,0.036,0.01,0.003,0.001,0.001,0,0,0,0
Agent5	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1	0.385,0.169,0.084,0.047,0.03,0.02,0.015,0.013,0.0

Fig. 3: Agent Guilt Probability Distribution

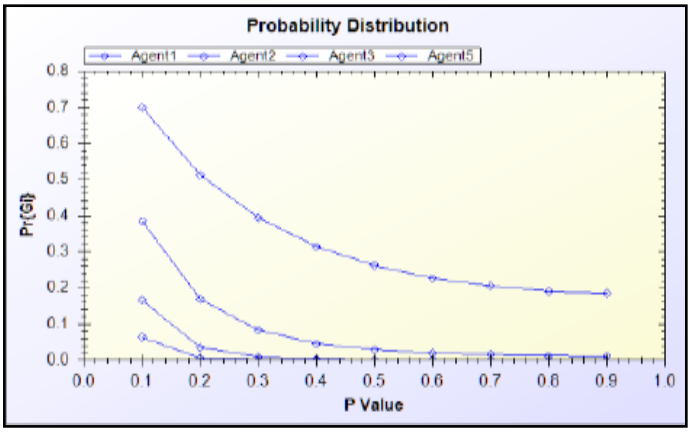

Fig. 4: Graphical Representation of Probability Distribution

In malice 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. The data distribution strategies improve the distributor's 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 —practical 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.

References

[1] Panagiotis Papadimitriou, Student Member, IEEE, "Hector Garcia-Molina, Member, IEEE‖ Data Leakage Detection IEEE Transactions On Knowledge And Data Engineering, Vol. 23, No. 1, January 2011.

[2] Cisco white paper on Data Leakage Worldwide: The High Cost of Insider Threats, 2008.

[3] Y. Cui, J. Widom,"Lineage Tracing for General Data Warehouse Transformations", The VLDB J., Vol. 12, pp. 41 58, 2003.

[4] P. Buneman, W.-C. Tan, In SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pp. 1171–1173, 2007.

[5] J.J.K.O. Ruanaidh, W.J. Dowling, F.M. Boland, -Watermarking Digital Images for Copyright Protection", IEEE Proc. Vision,

[6] Oztan Harmanci, M. Kivanc Mihcak, A. Murat Tekal-Watermarking and streaming compressed video", pp. 833-836 June 2007

[7] Panagiotis Papadimitriou,"Hector Garcia-Molina‖ A Model for Data Leakage Detection‖ Stanford University 353 Serra Street", Stanford, CA 94305, USA.

[8] R. Agrawal, J. Kiernan,"Watermarking Relational Databases", Proc. 28th Int'l Conf. Very Large Data Bases (VLDB '02), VLDB Endowment, pp. 155-166, 2002.

[9] P. Buneman, W.-C. Tan,"Provenance in Databases, Proc. ACM SIGMOD, pp. 1171-1173, 2009.

[10] Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.



Karpurapu Sudhakara Babu received his POST Gradution degree in MCA from, Indira Gandhi National Open Unversity New Delhi in 2005, the M.TECH. Degree in CSE from Lenora College Of Engineering, in 2012. At present, He is engaged in" Data Leakage Detection.



Pendurthi Sudhakar Babu received the M Tech degree from JNTU.K, Jawarharlal Nehru Technological University, and Kakinada in 2012. Currently he is working as Asst.Professor Lenora Collegeof Engineering, Rampachodavarm East Godarvari Dist. in Andhra Pradesh, India.