Global Journal of Computer Science and Technology
Software & Data Engineering
Volume 13 Issue 6 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Data Leakage Detection by using Fake Objects

By Rama Rajeswari Mulukutla & P. Poturaju
Grandhi Varalakshmi VenkataRao Institute of Technology, India

Abstract - Modern business activities rely on extensive email exchange. Email leakage have became widespread throughout the world, and severe damage has been caused by these leakages it constitutes a problem for organization. We study the following problem: A data distributor has given sensitive data to a set of supposedly trusted agents (third parties).If the data distributed to the third parties is found in a public\private domain then finding the guilty party is a nontrivial task to a distributor. Traditionally, this leakage of data has handled by water marking technique which requires modification of data. If the watermarked copy is found at Some unauthorized site then distributor claim his ownership. To overcome the disadvantage of using watermark, data allocation strategies are used to improve the probability of identifying guilty third parties. The distributor must assess the likelihood that the leaked data come from one or more agents, as opposed to having been gathered from other means. In this project, we implement and analyze a guilt model that detects the agents using allocation strategies without modifying the original data .the guilt agent is one who leaks a portion of distributed data. We propose data "realistic but fake" data records to further improve our chances of detecting leakage and identifying the guilty party. And Algorithms implemented using fake objects will improve the distributor chance of detecting the guilt agent. It is observed that by minimizing the sum objective the chance of detecting guilt agents will increase. We also develop a framework for generating fake objects.

Keywords : *allocation strategies, data leakage, data privacy, fake records, leakage model.*

GJCST-C Classification : *H.2.m*

Strictly as per the compliance and regulations of:

Data Leakage Detection by using Fake Objects

Rama Rajeswari Mulukutla [α] & P. Poturaju [σ]

Abstract - Modern business activities rely on extensive email exchange. Email leakage have became widespread throughout the world, and severe damage has been caused by these leakages it constitutes a problem for organization. We study the following problem: A data distributor has given sensitive data to a set of supposedly trusted agents (third parties).If the data distributed to the third parties is found in a public\private domain then finding the guilty party is a nontrivial task to a distributor. Traditionally, this leakage of data has handled by water marking technique which requires modification of data. If the watermarked copy is found at Some unauthorized site then distributor claim his ownership. To overcome the disadvantage of using watermark, data allocation strategies are used to improve the probability of identifying guilty third parties. The distributor must assess the likelihood that the leaked data come from one or more agents, as opposed to having been gathered from other means. In this project, we implement and analyze a guilt model that detects the agents using allocation strategies without modifying the original data .the guilt agent is one who leaks a portion of distributed data. We propose data "realistic but fake" data records to further improve our chances of detecting leakage and identifying the guilty party. And Algorithms implemented using fake objects will improve the distributor chance of detecting the guilt agent. It is observed that by minimizing the sum objective the chance of detecting guilt agents will increase. We also develop a framework for generating fake objects.

Keywords : allocation strategies, data leakage, data privacy, fake records, leakage model.

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 the owner of the data the distributor and the supposedly trusted third parties the agents. Our goal is to detect when the distributor's sensitive data have been leaked by agents, and if possible to identify the agent that leaked the data.

According to 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 referred as business process outsourcing (BPO) and it allow the companies to focus on their core competency by subcontracting with other activities to specialists, resulting in reducing the operational costs ,and increasing the productivity. Security and business assurance are essential for BPO.

In many cases the service provider needs access to the company 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 (social security numbers),a patenting law firm to some research results, a marketing service vendor to contact information for customers or a payment service provider may need to access the credit card numbers or bank account numbers.

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 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 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 website, 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.

II. Problem Definition

Suppose a distributor owns a set $T = \{t_1, t_m\}$ of valuable data objects. The distributor wants to share some of the objects with a set of agents $U_1, U_2, \ldots, U_n$. An agent U_i receives a subset of objects R_i which belongs to T, determined either by a sample request or

Author α : M.Tech(CSE), Department of Computer Science & Engineering, Grandhi Varalakshmi VenkataRao Institute of Technology Affiliated to JNTUK. E-mail : rama_mulukutla2003@yahoo.com
Author σ : Associate Professor, M.Tech, Department of Computer Science & Engineering, Grandhi Varalakshmi VenkataRao Institute of Technology Affiliated to JNTUK. E-mail : raju1poturaju1@gmail.com

an explicit request, Sample request R_i = SAMPLE(T, m_i): Any subset of m_i records from T can be given to Ui. Explicit request Ri = EXPLICIT (T, $cond_i$): Agent U_i receives all T objects that satisfy condi. The objects in T could be of any type and size, e.g., they could be tuples of relation, or relations in a database. After giving objects to agents, the distributor discovers that a set S of T has leaked. This means that some third party, called the target, has been caught in possession of s. For example ,this target may be displaying Son its website ,or perhaps as part of a legal discovery process, the target turned over s to the distributor Since the agents $U_1, \ldots, U_n$ have some of the data, it is reasonable to suspect them leaking the data. However, the agents can argue that they are innocent, and that the S data were obtained by the target through other means

a) Agent Guilt Model

Suppose an agent U_i is guilty if it contributes one or more objects to the target. The event that agent U_i is guilty for a given leaked set S is denoted by $G_i|S$. The next step is to estimate $Pr\{G_i|S\}$, i.e ., the probability that agent Gi is guilty Given evidence S. To compute $Pr\{G_i|S\}$, estimate the probability that values in S can be "guessed" by the target.

For instance, say that some of the objects in t are e mails of individuals. Conduct an experiment and ask a person with approximately the expertise and resources of the target to find the e-mail of, say, 100 individuals, the person may only discover 20, leading to an estimate of 0.2. We call this estimate pt, the probability that object t can be guessed by the target.

The two assumptions regarding the relationship among the various leakage events. Assumption 1. For all t, t ϵS such that' t $\neq$t the provenance of t is independent of the provenance of t′. The term "provenance" in this assumption statement refers to the source of a value t that appears in the leaked set. The source can be any of the agents who have t in their sets or the target itself (guessing). Assumption 2. An object t ϵ S can only be obtained by the target in one of the two ways: A single agent U_i leaked t from its own R_i set. The target guessed (or obtained through other means) t without the help of any of the n agents.

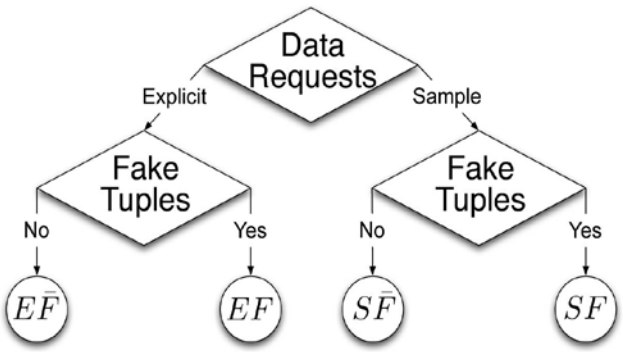
To find the probability that an agent Ui is guilty, given a set S, Consider. The target guessed t1 with probability p. and that agent leakes t1 to S with probability 1 –p. First compute the probability that he leaks a single object t to S. To compute this, define the set of agents V_t = $\{U_i | t \epsilon R_i\}$ that have t in their data sets. Then using assumption2 and known probability p, we have Pr{some agent leaked t to S}=1-P……(.1.1). Assuming that all agents that belong to V_t can leak t to S with equal probability and using Assumption 2we obtain, Pr{ U_i leaked t to S}={1-p/$|V_t|$, 0, if $U_i \epsilon V_t$, o……(1.2) Otherwise. Given that agent U_i is guilty if he leaks at least one value to S, with Assumption 1 and equation 1.2 compute the probability $Pr\{G_1|S\}$, agent U_i is guilty, $Pr\{G_1|S\}=1-\pi_{t\epsilon cs\cap Ri}(1-p/|_{vt|})$.

b) Data Allocation Problem

The distributor "intelligently" gives data to agents in order to improve the chances of detecting a guilty agent. There are four instances of this problem, depending on the type of data requests made by agents and whether "fake objects" are allowed.

Agent makes two types of requests, called sample and explicit. Based on the request the Fake objects are added to the data list. Fake objects are objects generated by the distributor that are not in set T. The objects are designed to look like real objects, and are distributed to agents together with T objects, in order to increase the chances of detecting agents that leak data.

Figure 1 : Leakage Instance Problems



The Figure represents our four problem instances with the names EF, EF, SF, and SF, where E stands for explicit requests, S for sample requests, F for the use of fake objects, and F for the case where fake objects are not allowed.

The distributor may be able to add fake objects to the distributed data in order to improve in his effectiveness in detecting guilty agents. Since, fake objects may impact the correctness of what agents do, So they may not be allowable. Use of fake objects may be inspired by the use of "trace" records in mailing lists. The distributor creates and add fake objects to the that he distributes to the agents .In many cases, the distributor may be limited how many fake objects he can create.

In EF problems, objectives values are initialized by agent's data requests. Say for example, that t= $\{t_1,t_2\}$ and $R_2=\{t_1\}$. The distributor cannot remove or alter the R_1 or R_2 to the data to decrease the overlap $R_1\backslash R_2$. However, say the distributor can create one fake object (B=1) and both agents can receive one fake objects($b_1=b_2=1$).If the distributor is able to create fake objects, he could improve further objective.

III. Optimization Problem

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. The fake object distribution as the only possible constraint relaxation. The objective is to maximize the chances of detecting a guilty agent that leaks all its data objects. The $Pr\{G_j|S = R_i\}$ or simply $Pr\{G_j|R_i\}$ is the probability that agent Uj is guilty if the distributor discovers a leaked table S that contains all Ri objects. We define the difference functions $\Delta(I,j)$is defined as $\Delta(i; j) = Pr\{G_i|R_i\} - Pr\{G_j|R_i\}$.

a) Problem Definition

Let the distributor have data request from n agents .The wants to give tables $R_1 \ldots\ldots R_n$ to agents $U_1 . . . U_n$ respectively, so that Distributor satisfies agent's requests; and Maximizes the guilt probability differences $\Delta(i,j)$ for all i,j=1…..n and i=j. Assuming that the R1 sets satisfy the agent's requests, we can express the problem as a multi–criterion.

b) Optimization Problem

Maximize(…,Δ(I,j),…)i!=j…….(1.5) (over $R_1,\ldots,R_n$). The approximation [3] of objective of the above equation does not depend on the agent probabilities and therefore minimize there altive overlap among the agents as Minimize(…,|Ri∩Rjj/ Ri,…..)i!=j…….(1.6) over($R_1,\ldots,R_n$). This approximation valid if minimizing the relative overlap, $|Ri \cap Rj| \backslash R_i$ maximizes(i,j).

IV. Objective Approximation

In case of sample request, all request are fixed size. Therefore, maximize the chance of detecting a guilt agent that leaks all his data by minimizing,| Ri ∩Rj|\Ri is equivalent to minimizing |Ri ∩Rj|. The minimum value of |Ri ∩ Rj| maximizes $\prod$ |Ri ∩Rj| and Δ(i,j) since π |Ri | is fixed. If agents have explicit data requests, that overlaps |Ri ∩Rj| are defined by their own requests and |Ri ∩Rj| are fixed .Therefore minimizing |Ri| j is equivalent to maximizing |Ri|(with the addition of fake objects).The maximium value of |Ri| minimizes$\prod$ |Ri| and maximizes Δ(i,j), since $\prod$ (Ri ∩Rj) is fixed. Our paper focus on identifying the leaker .So we propose to trace the ip address of the leaker. The file is send to the agents in the form of email attachments which need a secret key to download it. This secret key is used to generate random function and send to the agent either on the mobile number used at registration or to the other global email service such as gmail. Whenever secret key mismatch takes place the fake file gets downloaded. To further enhance our objective approximation ip address tracking is done of the system where fake object is downloaded .Various commands are available for getting ip address information. ping, tracert etc may one be used to get it. The ip address traced with time so as to overcome problem of dynamic ip addressing .But as we are doing in organization there is no problem of dynamic ip. Or else looking for the ip address universally it is unique that period of time therefore it can be traced to the unique system of the leaker.

V. Allocation Strategies

In this section, the allocation strategies that solve exactly or approximately the scalar versions of equation 1.7 for the different instances presented in Fig.1. In this Section. A deals with problems with explicit data requests, and in Section B with problems with sample data requests.

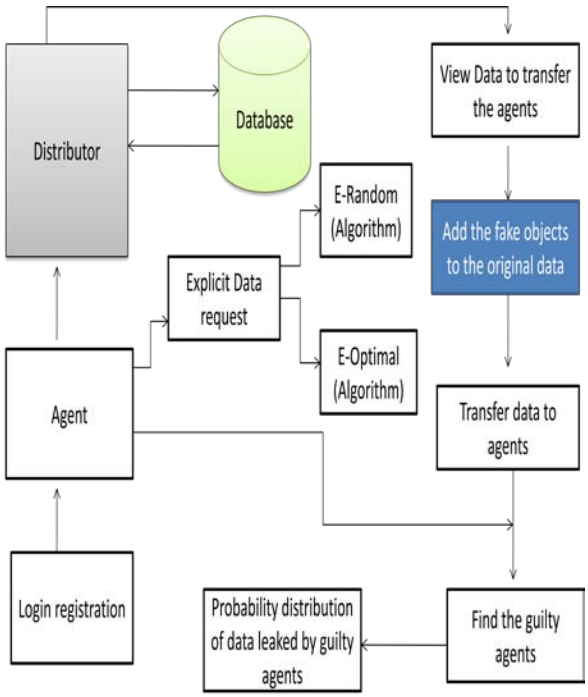


Figure 2 : Architecture of Distributor

a) Explicit Data Requests

In case of explicit data request with fake not allowed problem to add fake objects to the distributed data. So, the data allocation is fully defined by the agents' data requests. In case of explicit data request with fake allowed, the distributor cannot remove or alter the request R from the agent .However distributor can add fake object. In algorithm for data allocation for explicit request, the input to this is a set of request $R_1,\ldots\ldots, R_n$ from n agents and different conditions for requests. The e-optimal algorithm finds the agent that is

eligible to receiving fake objects. Then create one fake object for iteration and allocate it to the agent selected. The e-optimal algorithm minimizes every term of the objective summation by adding maximum number bi of fake objects to every set Ri yielding optimal solution.

Step 1: Calculate total fake records as sum of fake records allowed.
Step 2: While total fake objects >0.
Step 3: Select the agent that yield the greatest improvement in sum objective i.e;i=$^{argmax}(1\backslash|R1|-1\backslash|R1|+1)\Sigma_j: {}_{Ri}\cap_{Rj}$.
Step 4: Create fake record.
Step 5: Add this fake record to the agent and also to fake record set.

b) Sample Data Requests

With sample data requests, each agent Ui may receive any T subset out of ($|T|_m$) different ones. Hence, there are $\prod^n_{i=1}(|T|\ m)$ 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.

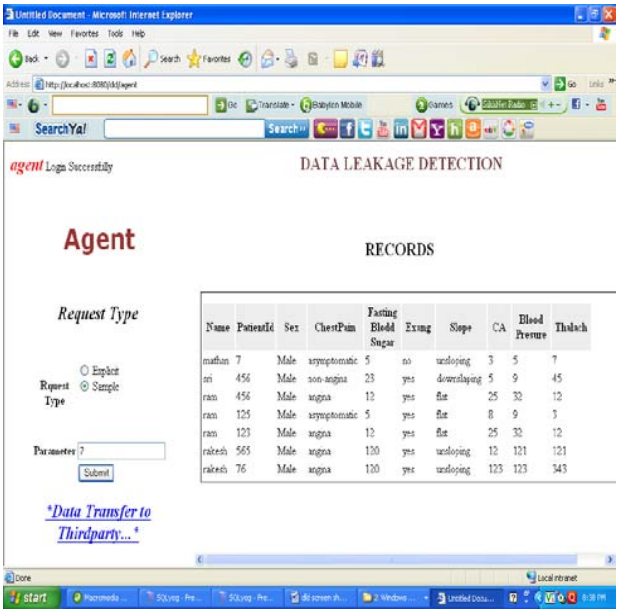
Therefore, from the distributor's perspective, there are $\prod^n_{i=1}(|T|_m)/|T|$ different allocations. An object allocation that satisfies requests and ignores the distributor's objective is to give each agent a unique subset of size m. The s-max algorithm allocates to an agent the data record that yields the minimum increase of the maximum relative overlap among any pair of agents. The s- max algorithm as follows:

Step 1: Initiliaze Min_overlap ←1,the minimum out of the maximum relative overlaps that the allocation of different objects to Uj.
Step 2: For k ϵ{k′|$t_k \epsilon$Ri } do.
Step 3: For all j=1,…n: j=1 and $t_k \epsilon_{Rj}$ do.

Calculate absolute overlap as
abs_ov←|Ri ∩Rj|+1

Calculate relative overlap as:

rel_ov←abs_ov/min(m_i,m_j)
Step4: Find maximum relative as
max_rel_ov←Max(max_rel_ov,rel_ov)
If max_rel_ov ≤ min_overlap then
min_overlap←max_rel_ov
ret_k←k
Return ret_k



VI. Related Work

The guilt detection approach we present is related to the data provenance problem tracing the lineage o f S objects implies essentially the detection of the guilty agents Suggested solutions are domain specific, such as lineage tracing for data warehouses, 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 and simplifies lineage tracing, since we do not consider any data transformation from Ri sets to S.

As far as the data allocation strategies are concerned, our work is mostly relevant to watermarking that is used as a means of establishing original ownership of distributed objects. Watermarks were initially used in images video and audio data whose digital representation includes considerable redundancy. 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. Finally, there are also lots of other works on mechanisms that allow only authorized users to access sensitive data through access sensitive data 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 agents' requests.

VII. Conclusion and Future Work

In spite of these difficulties, we have shown 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 probability that objects can be "guessed" by other means. Our model is relatively simple, but we believe that it captures the essential trade-offs. The algorithms we have presented implement a variety of data distribution strategies that can improve the distributor's chances of identifying a leaker.

We 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. Our future work includes the investigation of agent guilt models that capture the leakages.

Acknowledgement

I would like to express my sincere thanks to my guide and my authors for their consistence support and valuable suggestions.

References Références Referencias

1. R. Agrawal and J. kiernan, "Watermarking Relational Databases", proc.28th Intl conf. very Large Data Bases (VLDB'02), VLDB Endowment, pp, 155-166, 2002.
2. 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.
3. P. Buneman and W.C.T an, "Provenance in Databases", proc.ACM SIGMOD, pp. 1171-1173, 2007.
4. F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video", Signal processing, vol.66, no.3, pp.283-30, 1998.
5. B. Mungamuru and H. Garcia-Molina, "Privacy, Preservation and Performance: The 3 P's of Distributed Data Management", technical report, Stanford Univ., 2008.
6. P. Papadimitriou and H. Garcia–Molina, "Data Leakage Detection", technical report Stanford Univ., 2008.
7. L. Sweeney "Achieving K-Anonymity Privacy Protection Using Generalization and Suppression", http://en.scientificcommons.org/4319613, 2002.
8. Y. Li, V. Swarup and S. Jajodia, "Fingerprinting Relational Databases: Schemes and Specialties", IEEE trans. Dependable and Secure Computing, vol.2, pp.33-45, jan-mar. 2005.