Data Security and Robustness

Girish Reddy Ginni[1*], V.Suresh[2*], Vennakula I S Saikumar[3*]
1. M.Tech (CSE) Student, Dept of CSE, Pydah College of Engg & Tech, Vishakapatnam.
2. Assoc Professor, Dept of CSE, Pydah College of Engg & Tech, Vishakapatnam.
3. M.Tech (CSE) Student, Dept of CSE, Pydah College of Engg & Tech, Vishakapatnam.

Keywords:
Sensitive data, fake objects; data allocation strategies

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. 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.*

1. INTRODUCTION

In today technically empowered data rich environment, it is a major challenge for data holders to prevent data leakage. Loss of large volumes of protected information has become regular headline event, forcing companies to re-issue cards, notify customers and mitigate loss of goodwill from negative publicity.

While great deal of attention has been given to protecting companies electronic assets from outsiders threats – from intrusion prevention systems to firewalls to vulnerability management – organizations now turn their attention to an equally dangerous situation: the problem of data loss from the inside. Whether its email, instant messaging, webmail, a form of website, or a file transfer, electronic communications exiting the company still go largely uncontrolled and unmonitored on their way to their destinations – with the ever present potential for confidential information to fall into wrong hands. Should sensitive information be exposed, it can wreak havoc on the organization’s bottom line through fines, bad publicity, loss of strategic customers, loss of competitive intelligence and legal actions.

Consider the example where a former employee of one company accidentally post IDs and bank accounts data for 150 employees of an advertising firm on a website. The list goes on and on.

There is major solution given is “watermarking” technique, where the unique code is embedded within the data. But it is not useful with sensitive information as it changes some of bits in data. Also if the recipient is malicious it, may destroy the watermark.

Also access control mechanism can be used, that allow only authorized users to access the sensitive data through an access control policies. But it also put restrictions on users and our aim is to provide service to all customers (you cannot deny the coming request).

In this paper, we proposed one model that can handle all the requests from customers and there is no limit on number of customers. The model gives the data allocation strategies featured with the forged objects injection proposed by Ref [6] to improve the probability of identifying leakages, but they can accept request from only some number of customers.

***** **Girish Reddy Ginni**
M.Tech (CSE) Student, Dept of CSE, Pydah College of Engg & Tech, Vishakapatnam.

Also we study the application where there is a distributor, distributing and managing the files that contain sensitive information to users when they send request. The log is maintained for every request, which is later used to find overlapping with the leaked file set and the subjective risk assessment of guilt probability.

2. RELATED WORK

Here I proposed a watermarking algorithm that embeds the watermark bits in the least signi.cant bits (LSB) of selected attributes of a selected subset of tuples.This 3 technique does not provide a mechanism for multibit watermarks; instead only a secret key is used. For each tuple, a secure message authenticated code (MAC) is computed using the secret key and the tuple•s primary key. The computed MAC is used to select candidate tuples, attributes and the LSB position in the selected attributes. Hiding bits in LSB is efficient. However, the watermark can be easily compromised by very trivial attacks. For example a simple manipulation of the data by shifting the LSB•s one position easily leads to watermark loss without much damage to the data. Therefore the LSB-based data hiding technique is not resilient. Moreover, it assumes that the LSB bits in any tuple can be altered without checking data constraints. Simple unconstrained LSB manipulations can easily ggenerate undesirable results such as changing the age from 20 to 21.IN This We have presented a technique for fingerprinting relational data by extending watermarking scheme. a watermarking technique that embeds watermark bits in the data statistics. The data partitioning technique used is based on the use of special marker tuples which makes it vulnerable to watermark synchronization errors resulting from tuple deletion and tuple insertion; thus such technique is not resilient to deletion and insertion attacks.

The data manipulation technique used to change the data statistics does not systematically investigate the feasible region; instead a naive unstructured technique is used which does not make use of the feasible alterations that could be performed on the data without affecting its usability. Furthermore,a threshold technique for bit decoding that is based on two thresholds. However, the thresholds are arbitrarily chosen without any optimality criteria. Thus the decoding algorithm exhibits errors resulting from the non-optimal threshold selection, even in the absence of an attacker.

3. 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 sys- tem more focus on data allocation problem.

4. PROPOSED SYSTEM

To find the solution on this 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, Bambhori, and Jalgaon 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

4.1. 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].

4.2. 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.

4.3. Guilt Assessment

Let L denote the leaked data set that may be leaked intentionally or guessed by the target user.

Since agent having some of the leaked data of L, may be susceptible for leaking the data. But he may argue that he is innocent and that the L data were obtained by target through some other means.

Our goal is to assess the likelihood that the leaked data came from the agents as opposed to other resources.e.g. if one of the object of L was given to only agent A1, we may suspect A1 more. So probability that agent A1 is guilty for leaking data set L is denoted as Pr {Gi | L}.

4.4. Guilt Probability Computation

For the sake of simplicity our model relies on two assumptions:

Assumption 1: For all t_1, t_2, ………., t_n Є L and $t_1 \neq t_2$, the provenance of $t_1$is independent of t_2 .

Assumption 2: Tuple tЄL can only be obtained by third user in one of the two ways:

1. Single user A_1 leaked t or
2. Third user guessed t with the help of

other resources.

Now to compute the guilt probability that he leaks a single object t to L, we define a set of users.

To find the probability that an agent A_i is guilty for the given set L, consider the target guessed t_1 with probability p and that agent leaks t_1 to L with probability 1-p. First compute the probability that he leaks a single object to L. To compute this, define the set of agents $U_t = \{ A_i \mid t Є R_i \}$ that have t in their data sets. Then using Assumption 2 and known probability p,we have,

$$\Pr\{\text{Some agent leaked t to L}=1-p \quad (1)$$

Assuming that all agents that belongs to U_t can leak t to L with equal probability and using Assumption 2 we get,

$$Pr(\text{Ai leaked t to L}) = \begin{cases} \frac{1-p}{U_t}, & if\ A_i \in U_t \\ 0 & Otherwise \end{cases} \quad (2)$$

Given that user Ai is guilty if he leaks at least one value to L, with assumption 1 and equation 2, we can compute the probability Pr {Gi | L} that user Ai is guilty:

$$\Pr\{\text{Gi} \mid \text{L}\} = 1 - \prod_{t \in L \cap Ri} \left(\frac{1-(1-p)}{Ut} \right)$$

4.5. Data Allocation Strategies

The distributor gives the data to agents such that he can easily detect the guilty agent in case of leakage of data. To improve the chances of detecting guilty agent, he injects fake objects into the distributed dataset. These fake objects are created in such a manner that, agent cannot distinguish it from original objects. One can maintain the separate dataset of fake objects or can create it on demand. In this paper we have used the dataset of fake tuples.

Depending upon the addition of fake tuples into the agent's request, data allocation problem is divided into four cases as:

i. Explicit request with fake tuples
ii. Explicit request without fake tuples
iii. Implicit request with fake tuples
iv. Implicit request without fake tuples.

For example, distributor sends the tuples to agents A_1 and A_2 as R_1= {t_1, t_2} and R_2= { t_1}. If the leaked dataset is L={ t_1}, then agent A_2 appears more guilty than A_1. So to minimize the overlap, we insert the fake objects in to one of the agent's dataset.

4.6. Overlap Minimization

The distributor's data allocation to agents one constraint and one objective. The distributor's constraint is to satisfy agent's request, 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 his constraint as strict. The distributor may not deny serving an agent request and may not provide agents different perturbed versions of the same object.

The objective is to maximize the chances of detecting guilty agent that leaks all his data objects.

The Pr {G_i | L=Ri } is the probability that agent A_i is guilty if distributor discovers a leaked table L that contains all Ri objects.

The difference function Δ (i, j) is defined as

$$\Delta(i,j) = \Pr\{G_i|R_i\} - \Pr\{G_j|R_i\} \qquad i,j = 1,\dots\dots.n$$

Problem definition: Let the distributor have data request from n agents. The distributor wants to give tables R_1, R2........ R_n to agents A_1, A2............. A_n respectively, so that

- Distribution satisfies agent's request; and
- Maximizes the guilt probability differences Δ (i, j) for all i, j= 1, 2, ……n and i≠j.

Optimization Problem:

Maximizing the difference among distributed dataset increases the minimization of overlap.

i.e $\max_{(over\ R_1,\dots\dots\dots,R_n)}(\dots\dots,\Delta(i,j),\dots.) \qquad i\neq j$

Then

$$\min_{(over\ R_1,\dots\dots,R_n)}\left(\dots\dots,\left|\frac{R_i\cap R_j}{|R_i|}\right|\right) \qquad i\neq j$$

5. EXPERIMENTAL SETUP

In this paper, we presented the algorithm and the corresponding results for the explicit data allocation with the addition of fake tuples. We are still working on minimizing the overlap in case of implicit request. Whenever any user request for the tuple, it follows the following steps:

1. The request is sent by the user to the distributor.
2. The request may be implicit or explicit.
3. If it is implicit a subset of the data is given.
4. If request is explicit, it is checked with the log, if any previous request is same .
5. If request is same then system gives the data objects that are not given to previous agent.
6. The fake objects are added to agent's request set.
7. Leaked data set L, obtained by distributor is given as an input.
8. Calculate the guilt probability G_i of user using II.

In the case where we get similar guilt probabilities of the agents, we consider the trust value of agent. These trust values are calculated from the historical behavior of agents. The calculation of trust value is not given here, we just assumed it. The agent having low trust value is considered as guilty agent.

The algorithm for allocation of dataset on agent's explicit request is given below.

5.1 Algorithm :

Allocation of Data Explicitly:

Input:-

i. $T=\{t_1,t_2,t_3,\ldots\ldots t_n\}$ -Distributor's Dataset
ii. R- Request of the agent
iii. Cond- Condition given by the agent
iv. m= number of tuples given to an agent m<n, selected randomly

Output:- D- Data sent to agent

```
1.  D=Φ, T'=Φ
2.  For i=1 to n do
3.  If($t_i$.fields==cond) then
4.  T'=T'U{ $t_i$}
5.  For i=0 to i<m do
6.  D=DU{ $t_i$}
7.  T'=T'-{ $t_i$}
8.  If T'=Φ then
9.  Goto step 2
10. Allocate dataset D to particular agent
11. Repeat the steps for every agent
```

To improve the chances of finding guilty agent we can also add the fake tuples to their data sets. Here we maintained the table for duplicate tuples and add randomly these tuples to the agent's dataset.

Algorithm2:

5.2 Addition of fake tuples:

Input:

i. D- Dataset of agent
ii. F- Set of fake tuples
iii. Cond - Condition given by agent iv.b- number of fake objects to be sent

Output:- D- Dataset with fake tuples

```
1. While b>0 do
2. f= select Fake Object at random from set F
3. D= DU {f}
4. F= F-{f}
5. b=b-1
6. if F=Φ then reinitialize the fake data set.
```

Similarly, we can distribute the dataset for implicit request of agent. For implicit request the subset of distributor's dataset is selected randomly. Thus with the implicit data request we get different subsets. Hence there are different data allocations. An object allocation that satisfies requests and ignores the distributor's objective to give each agent unique subset of T 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 is as follows:

```
1. Initialize Min_Overlap, the minimum out of the
   minimum relative overlaps that the allocations of
   different objects to $A_i$
2. for k do Initialize max_rel_ov←0, the maximum
   relative overlap between $R_i$ the allocation of $t_k$ to $A_i$
3. for all j=1,……,n:j=I and tk ЄRj do calculate absolute
   overlap as abs_ov← calculate relative overlap as
   rel_ov←abs_ov/min(mi, mj)
4. Find maximum relative overlap as
   Max_rel_ov←MAX(max_rel_ov, rel_ov)

     If max_rel_ov≤ min_ov then
     Min_ov←max_rel_ov
     ret_k←k
     Return ret_k
```

The algorithm presented implements a variety of data distribution strategies that can improve the distributor's chances of identifying a leaker. It is 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.

6. CONCLUSION

In this paper 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 .

REFERENCES

1. YIN Fan, WANG Yu, WANG Lina, Yu Rongwei. A Trustworthiness-Based Distribution Model for Data Leakage Detection: Wuhan University Journal Of Natural Sciences.

2. P. Buneman, S. Khanna and W.C. Tan. Why and where: A characterization of data provenance. ICDT 2001, 8th International Conference, London, UK, January4-6, 2001, Proceedings, volume 1973 of Lecture Notes in Computer Science, Springer, 2001.

3. S. Czerwinski, R. Fromm, and T. Hodes. Digital music distribution and audio watermarking.

4. Rakesh Agrawal, Jerry Kiernan. Watermarking Relational Databases// IBM Almaden Research Center

5. S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian. Flexible support for multiple access control policies. ACM Trans. Dataset Systems, 26(2):214-260,2001.

6. Papadimitriou P, Garcia-Molina H. A Model For Data Leakage Detection// IEEE Transaction On Knowledge And Data Engineering Jan.2011.

7. L. Sweeney. Achieving k- anonymity privacy protection using generalization and suppression. International Journal on Uncertainty, Fuzzyness and Knowledge-based Systems-2002

8. K.Suresh ,R.MadanaMohana and Dr.A.RamaMohan Reddy “crime analysis using data mining”, IJEECT International Journal of Electrical ,Electronics and Computing Technology, Vol.1(2), Jan-April, 2011, pp 58-63 ,ISSN 2229-3027.