ISSN:0975-9646

Unobtrusive Technique for the Detection of Data Seepage

Anusha.Koneru,G.Siva Nageswara Rao,J.Venkata Rao
Department of CSE, K L University
Guntur, Andhra Pradesh, India

ABSTRACT-Data leakage is a budding security threat to organizations, particularly when data leakage is carried out by trusted agents. In this paper, we present unobtrusive techniques for detecting data leakage and assessing the "guilt" of agents. Water marking is the long-established technique used for data leakage detection which involves some modification to the original data. To overcome the disadvantages of using watermark, data allocation strategies are used to improve the feasibility of detecting guilty agent. Distributor "intelligently" allocates data based on sample request and explicit request using allocation strategies in order to better the effectiveness in detecting guilty agent. Fake objects are designed to look like real objects, and are distributed to agents together with requested data. 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.

Keywords:-Allocation strategies, data leakage, fake objects, guilty agent, optimization.

1.INTRODUCTION:

Data Leakage can occur through a variety of methods - some are simple, some complex. As such, there is no single "silver bullet" to control Data Leakage. Data leakage detection is an increasingly important part of any organization's ability to manage and protect critical and confidential information. Examples of critical and confidential data that applications can access include: Intellectual Property, Corporate Data, Customer Data. The goal of our paper is to detect when the distributor's sensitive data has been leaked by agents, and show the probability for identifying the agent that leaked the data.

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 illegitimate 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 as shown in Figure.1 for assessing the "guilt" of agents by considering the option of adding "fake" objects to the distributed set. We also present algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker.

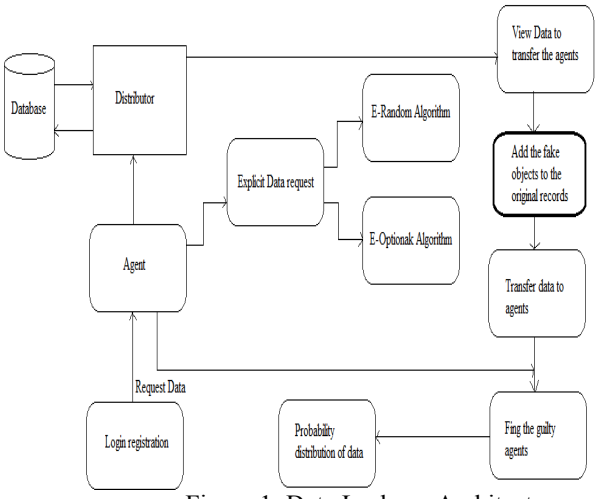

Figure.1: Data Leakage Architecture

2. 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 $U_1, U_2 \ldots U_n$ but does wish the objects be leaked to other third parties. An agent U_i receives a subset of R_i objects which belongs to *T*, determined either by a sample request or an explicit request,

- Sample Request R_i = SAMPLE (T, m_i): Any subset of *mi* records from *T* can be given to U_i.
- Explicit Request R_i = EXPLICIT (T, $cond_i$): Agent U_i receives all the *T* objects that satisfy $cond_i$.

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. 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 *S* on its web site, or perhaps as part of a legal discovery process, the target turned over *S* to the distributor. Since the agents $U_1, U_2, \ldots, U_n$ 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 was obtained by the target through other means.

2.1 . 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 $G_i \mid S$. The next step is to

estimate $P_r\{G_i| S\}$, i.e., the probability that agent G_i is guilty given evidence *S*.
To compute the $P_r\{G_i| S\}$, estimate the probability that values in *S* can be "guessed" by the target. For instance, say some of the objects in *t* are emails of individuals. Conduct an experiment and ask a person to find the email of say 100 individuals, the person may only discover say 20, leading to an estimate of 0.2. Call this estimate as p_t, 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*, $t \in S$ 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.

Assumption 2: An object $t \in S$ can only be obtained by the target in one of two ways.

- A single agent U_i leaked *t* from its own R_i set, or
- 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 leaks t_1 to *S* with the probability 1-*p*. First compute the probability that he leaks a single object *t* to *S*. To compute this, define the set of agents $V_t = \{U_i \mid t \in R_i\}$ that have *t* in their data sets. Then using Assumption 2 and known probability *p*, we have

$P_r\{\text{some agent leaked t to s}\}=1-p$ ……..1.1

Assuming that all agents that belong to V_t can leak *t* to *S* with equal probability and using Assumption 2 obtain,

$$p_r\{w_t \text{ leaked } t \text{ to } s\} = \begin{cases} \frac{1-p}{|V_t|}, & \text{if } w_t \in v_t \\ 0, & \text{otherwise} \end{cases} \quad \text{……1.2}$$

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 $P_r\{G_i \mid S\}$, agent U_i is guilty,

$$p_r\{G_i/s\} = 1 - \prod t \in s \cap R_i \left(1 - \frac{1-p}{|v_t|}\right) \quad \text{……1.3}$$

2.2. 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 requests the fakes objects are added to 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 the *T* objects, in order to increase the chances of detecting agents that leak data.

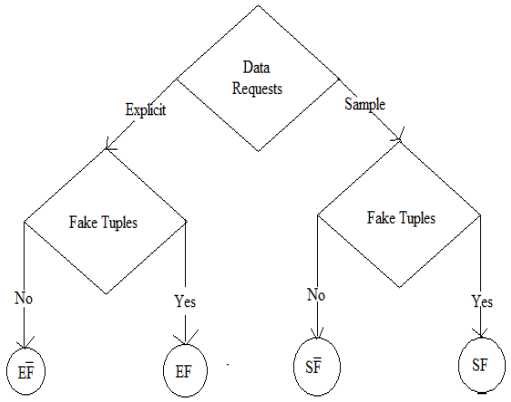

Figure. 2: Leakage Problem Instances

The Figure. 2 represents four problem instances with the names *EF*, *EF̄*, *SF* and *SF̄*, 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 his effectiveness in detecting guilty agents. Since, fake objects may impact the correctness of what agents do, so they may not always be allowable. Use of fake objects is inspired by the use of "trace" records in mailing lists. The distributor creates and adds fake objects to the data that he distributes to agents. In many cases, the distributor may be limited in how many fake objects he can create.
In *EF* problems, objective values are initialized by agents' data requests. Say, for example, that $T = \{t_1, t_2\}$ and there are two agents with explicit data requests such that $R_1 = \{t_1, t_2\}$ and $R_2 = \{t_1\}$. The distributor cannot remove or alter the R_1 or R_2 data to decrease the overlap $R_1 \setminus R_2$. However, say the distributor can create one fake object ($B = 1$) and both agents can receive one fake object ($b_1 = b_2 = 1$). If the distributor is able to create more fake objects, he could further improve the objective.

2.3. 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 his data objects. The Pr $\{G_j|S = R_i\}$ or simply Pr $\{G_j|R_i\}$ is the probability that agent U_j is guilty if the distributor discovers a leaked table *S* that contains all Ri objects.

The difference functions $\Delta(i, j)$ is defined as:

$$\Delta(i, j) = \Pr\{G_i \mid R_i\} - \Pr\{G_j \mid R_i\} \quad \ldots\ldots. 1.5$$

2.3.1 *Problem definition:* Let the distributor have data requests from *n* agents. The distributor wants to give tables $R_1, R_2 \ldots\ldots, R_n$ to agents $U_1, \ldots, U_n$, respectively, so that

- Distribution satisfies agents' requests; and
- Maximizes the guilt probability differences $\Delta(i, j)$ for all $i, j = 1 \ldots n$ and $i = j$.

Assuming that the Ri sets satisfy the agents' requests, we can express the problem as a multi-criterion

2.3.2 *Optimization problem:*

$$\text{maximize}_{(\text{over } R_1, \ldots, R_n)}(\ldots, \Delta(i,j), \ldots) \quad i \neq j \ldots\ldots 1.6$$

The approximation of objective of the above equation does not depend on agent's probabilities and therefore minimize the relative overlap among the agents as

$$\text{maximize}_{(\text{over } R_1, \ldots, R_n)}(\ldots, \frac{|R_i \cap R_j|}{|R_i|}, \ldots) \qquad i \neq j \ldots\ldots 1.7$$

This approximation is valid if minimizing these relative Overlap $\frac{|R_i \cap R_j|}{|R_i|}$ maximizes $\Delta(i, j)$.

2.4. Objective Approximation

In case of sample request, all requests are of fixed size.Therefore, maximizing the chance of detecting a guilty agent that leaks all his data by minimizing is $\frac{|R_i \cap R_j|}{|R_i|}$ equivalent to minimizing($|R_i \cap R_j|$).The minimum value of $|R_i \cap R_j|$. maximizes $\prod(|R_i \cap R_j|)$ and $\Delta(i,j)$, since $\prod(|R_i|)$ is fixed.If agents have explicit data requests, then overlaps($|R_i \cap R_j|$). are defined by their own requests and $|R_i \cap R_j|$ are fixed. Therefore, minimizing $|R_i|$ j is equivalent to maximizing |Ri | (with the addition of fake objects).The maximum value of $|R_i|$ minimizes $\Pi(R_i)$ and maximizes $\Delta(i, j)$,since $\prod(R_i \cap R_j)$ is fixed.

3. ALLOCATION STRATEGIES

3.1. Evaluation of Explicit Data Request Algorithms

In the first place, the goal of these experiments was to see whether fake objects in the distributed data sets yield significant improvement in our chances of detecting a guilty agent. In the second place, we wanted to evaluate our e-optimal algorithm relative to a random allocation.

3.2. Evaluation of Sample Data Request Algorithms

With sample data requests agents are not interested in particular objects. Hence, object sharing is not explicitly defined by their requests. The distributor is "forced" to allocate certain objects to multiple agents only if the number of requested objects exceeds the number of objects in set T. The more data objects the agents request in total, the more recipients on average an object has; and the more objects are shared among different agents, the more difficult it is to detect a guilty agent.

4.CONCLUSION:

Effective control of Data Leakage is multi-faceted. Identifying the probability of data leakage is dominant especially when the data is confidential and sensitive in nature. Usually leakage detection is handled by watermarking which modifies original objects before being transmitted for security reasons, our system does not need any alteration of original objects. We implemented various algorithms that are having different data allocation strategies meant for enhancing the probabilities of distributor in identifying the leaker. In this paper, we uses Fake data allocation schemes along with unobtrusive techniques like Allocation for Explicit Data Requests, Sample Data Request, Random Agent Selection, fake object allocation , Greedy Selection of agent for optimization, Data allocation, Object Selection etc were used to trace data leakages.

FUTURE WORK

Our future work includes the inquiring of agent guilt models that capture leakage scenarios that are not studied in this paper. For instance, 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).

ACKNOWLEDGMENT

We would like to thank Dr.V.Srikanth, Head of the Department, Computer Science & Engineering, KL university, Vaddeswaram for his encouragement and motivation to write this paper. Also we are grateful to G.Siva Nageswara Rao, (CSE), Kluniversity, Vaddeswaram for guiding us in writing this paper..

REFERENCES

[1] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression, 2002.

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

[3] P. Buneman and W.-C. Tan, "Provenance in Databases," *Proc*. ACM SIGMOD, pp. 1171-1173, 2007.

[4] F. Guo, J. Wang, Z. Zhang, X. Ye, and D. Li, "An Improved Algorithm to Watermark Numeric Relational Data," *Information Security Applications*, pp. 138-149, Springer, 2006.

[5] F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video," *Signal Processing*, vol. 66, no. 3, pp. 283-301, 1998.

[6]Y.Li,V. Swarup, and S. Jajodia, "Fingerprinting Relational Databases: Schemes and Specialties," *IEEE Trans. Dependable and Secure Computing,* vol. 2, no. 1, pp. 34-45, Jan.-Mar. 2005.

[7] B. Mungamuru and H. Garcia-Molina, "Privacy, Preservation and Performance: The 3 P's of Distributed Data Management," technical report, Stanford Univ., 2008.

[8] V.N. Murty, "Counting the Integer Solutions of a Linear Equation with Unit Coefficients," *Math. Magazine,* vol. 54, no. 2, pp. 79-81,1981.

[9] J.J.K.O. Ruanaidh, W.J. Dowling, and F.M. Boland, "Watermarking Digital Images for Copyright Protection,"*IEE Proc. Vision,Signal and Image Processing*, vol. 143, no. 4, pp. 250-256, 1996.

[10] R. Sion, M. Atallah, and S. Prabhakar, "Rights Protection for Relational Data," *Proc. ACM SIGMOD,* pp. 98-109, 2003.