DATA LEAKAGE DETECTION

Renuka Sabale **Supriya Chambule**

Eshaan Gupta **Vidya Waghmode**

Vishwakarma Institute Of Information Technology, Kondhwa(bk),pune-48,Maharastra,India.

Email:renuka.sable777@gmail.com,
supriyac567@gmail.com

ABSTRACT

The problem statement is –Distributor is giving data to trusted people called agents. And if we find the same data in an unauthorized place then distributor is checking if the data is leaked from agents or it is been gathered from some other means. If data is leaked by any of the agent ie if agent is found guilty then we have to find the guilty agent using data allocation strategies. For improving the chances of identifying guilty agent we are adding fake objects to the original data.

1 INTRODUCTION

While doing business, we have to share the data with other companies with which we are in partnership with or in case of hospitals patients records may be given to researchers to device new treatments. So we can say that sometimes we have to share the sensitive data. The one who is distributing data is called distributor and the trusted people whom we are distributing the data are called agents. Our goal is to detect if distributor's sensitive data is leaked by agents and to find out the guilty agent. Guilty agent is the one who is leaking data.

There are some applications where we need accurate data instead of perturbed one. Perturbation is the technique where data is made less sensitive. For example we can add range of values instead of exact values or we can add random noise to the data. But if outsourcer is doing payroll of a company then he should know the exact salaries and bank account numbers of all employees. Here we can't perturb the data.

Data leakage is handled by watermarking. In watermarking unique code is embedded in each distributor's copy. So if data leakage takes place then we can find out the guilty agent. Watermarks can be useful in some cases but they can be destroyed if agent is malicious.

In this paper we are studying the scenario where if set of objects are given to the agents and if same Objects found in any unauthorized place which may include any website or anyone's laptop then distributor is checking if that data is leaked from agent or the people who got the data got it from some other means. If distributor got enough evidence that an agent leaked the data then he may stop doing business with him.

While distributing the data distributor is adding some fake objects to the data, which will not be real but will look like realistic ones .So with the help of those fake objects distributor can find out guilty agent. Here fake objects will be different for every agent and they will work as watermark.

2 PROBLEM SETUP AND NOTATION

A distributor owns a set S={s1,s2,…sm} of valuable data objects. Distributor shares some of the objects with a set of agents A1,A2,….An but does not wish the objects be leaked to other third parties. The objects in S could be of any type and size. e.g. they could be tuples in a relation or relations in a database.

Any random agent Ai receives a subset of objects Ri determined by explicit data request as Ri=EXPLICIT (T, condi) : Agent Ai receives all S objects that satisfy condi.

Example - Say that S contains customer records for company C. Company C hires a marketing agency A1 to do an online survey of customers and company C subcontracts with agent A2 to handle billing for Calcutta agents. Thus A2 receives all S records that satisfy the condition "state is Calcutta"

3 GUILTY AGENTS

Suppose after giving objects to agent's distributor finds out that a subset Se from S has leaked. Means some data from S has been leaked and found on someone's website or perhaps as part of a legal discovery process.

As agents A1….An have that data,we can suspect them for leaking the data.But they can say that they are innocent and they got the data from some other means.e.g. The object from Se represents a customer X which can be a customer of some other company and that company provided that data.

Our goal is to find out if the data is leaked from the agents. If we found enough evidence then we can say that the agent is guilty e.g. more the data we found from a particular agent ,more we can suspect him.

We say agent Ai is guilty if it contributes one or more objects to the target then we can say that agent Ai is guilty by Gi and the event agent Ai is guilty for a given leaked set Se by Gi|Se. Our next step is to estimate Pr{Gi|Se} ,ie. The probability that agent Ai is guilty given by evidence Se.

4 FAKE OBJECTS

The distributor should be able to attach fake objects to the original data which is to be distributed to the third party agents in case to increase and improve the correctness in knowing or detecting the guilty agent who is leaking the data. However, the fake objects may impact the correctness of what agents do, so they may not always be allowable.

Actually the idea of adding a fake object or adding a state of equilibrium; especially a disturbance in the regular motion is not new, however in most cases, individual objects are perturbed, e.g., by adding noise to sensitive data records, or by hiding sensitive data behind an image or by adding watermarking by modifying some data so that, that data becomes less sensitive. In some applications these fake objects may cause some problems. For example, if it's a governments sensitive data even a small modification can cause a big mess another example, is of a hospital record in this case ,even a small modifications to the records of actual patients may be undesirable. Therefore we go for the concept of making fake objects, there is no harm in making the fake objects as fake objects are not real world entity objects so, considering the government example when such record does not exists than it wouldn't affect anyone or misguide anyone.

The use of these fake objects is similar like tracing the records in mailing lists. The distributor creates and adds fake objects to the data that he distributes to agents. We let Fi c Ri be the subset of fake objects that agent Ui receives. As discussed below, fake objects must be created carefully so that agents cannot distinguish them from real objects.

In almost all situations, the owner of the data may be in search of minimizing how many fake objects he can create. For example, objects may Contain e-mail addresses, and each fake e-mail address may require the creation of an actual inbox (otherwise, the agent May discover that the object is fake). The inboxes can actually be monitored by the distributor: if e-mail is received from someone other than the agent who was given the address, it is evident that the address was leaked. Since creating and monitoring e-mail accounts consumes resources, the distributor may have a limit of fake objects. If there is a limit, we denote it by B fake objects. Similarly, the distributor may want to limit the number of fake objects received by each agent so as to not arouse suspicions and to not adversely impact the agents' activities. Thus, we say that the distributor can send up to bi fake objects to agent Ui.

Creation of fake objects by creating of a fake object for agent Ui , and create a function CREATEOBJECT(Ri,Fi,condi) that takes as input the set of all objects Ri, the subset of fake objects Fi that Ui has received so far, and condi, and returns a new fake object. This function needs condi to produce a valid object that satisfies Ui's condition. Set Ri is needed as input so that the created fake object is not only valid but also indistinguishable from other real objects. CREATEOBJECT() function has to be aware of the fake objects added so far, again to ensure proper statistics.

5 ALLOCATION STRATEGIES

The algorithms to be applied are stated in the following section-

In case of class explicit request and use of fake objects, where E stands for explicit request and F stands for use of fake objects and f stands for fake objects not allowed. If the owner is able to create more fake objects, he Could further improve the objective. We present in Algorithms 1 and 2 a strategy for randomly allocating fake objects. Algorithm 1 is a general "driver" that will be used by other strategies, while Algorithm 2 actually performs the random selection. We denote the combination of Algorithm 1 with 2 as e-random. We use e-random as our baseline in our comparisons with other algorithms for explicit data requests.

Algorithm 1. Allocation for Explicit Data Requests (EF)

Input: R1; . . .;Rn, cond1; . . . ; condn, b1; . . . ; bn, B

Output: R1; . . .;Rn, F1; . . . ; Fn

```
1: R ; . Agents that can receive fake objects
2: for i = 1; . . . ; n do
3: if bi > 0 then
4: R ← R u { i }
5: Fi ← Ɵ
6: while B > 0 do
7: i ← SELECTAGENT(R;R1; . . .;Rn)
8: f ←CREATEFAKEOBJECT(Ri; Fi; condi)
9: Ri ← Ri
10: Fi ← Fi { f }
11: b← bi − 1
12: if bi = 0then
13: R ← R \ { Ri }
14: B← B −1
```

Algorithm 2. Agent Selection for e-random

```
1: function SELECTAGENT (R;R1; . . .;Rn)
2: i select at random an agent from R
3: return i
```

In lines 1-5, Algorithm 1 finds agents that are eligible to receiving fake objects in ^{O}(n) time. Then, in the main loop

in lines 6-14, the algorithm creates one fake object in every iteration and allocates it to random agent. The main loop

takes O(B) time. Hence, the running time of the algorithm is O(n + B). If B ≥ $\sum_{i=1}^{n} bi$,the algorithm minimizes every term of the objective summation by adding the maximum number bi of fake objects to every set Ri, yielding the optimal solution. Otherwise, if B <$\sum_{i=1}^{n} bi$ (as in our example where B = 1 < b1 + b2 = 2), the algorithm just selects at random

the agents that are provided with fake objects. We return back to our example and see how the objective would change if the distributor adds fake object f to R2 instead of R1. In this case, the sum-objective would be $\frac{1}{2} + \frac{1}{2}$ = 1 < 1:33.

The reason why we got a greater improvement is that the addition of a fake object to R2 has greater impact on the

Corresponding summation terms, since

$$\frac{1}{|R|} - \frac{1}{|R|+1} = \frac{1}{6} < \frac{1}{|R2|} - \frac{1}{R2+1} = \frac{1}{2}.$$

The left-hand side of the inequality corresponds to the objective improvement after the addition of a fake object to

R1 and the right-hand side to R2.

6 CONCLUSION

In this paper, we have presented an overview of Data Leakage detection through watermarking schemas. Watermarking provide security using various algorithms through encryption. We have discussed many of the current commercial and academic efforts to solve the problem of protecting copyrighted material. From this study, we conclude that the data leakage detection industry is very heterogeneous as it evolved out of ripe product lines of leading IT security vendors. A new technology such as firewalls, encryption, access control, identity management, machine learning content/context-based detectors and others have already been incorporated to offer protection against various facets of the data leakage threat. The competitive benefits of developing our System is to provide the highest degree of protection by ensuring an optimal fit of specific data leakage detection technologies also consider, the option of adding “fake” objects to the distributed set. Such objects do not correspond to real entities but appear realistic to the agents. In a sense, the fake objects act as a type of watermark for the entire set, without modifying any individual members. If it turns out that an agent was given one or more fake objects that Ire leaked, then the distributor can be more confident that agent was guilty.

REFERENCES

[1] 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.
[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, S. Khanna, and W.C. Tan, “Why and Where: A Characterization of Data Provenance,” Proc. Eighth Int’l Conf. Database Theory (ICDT ’01), J.V. den Bussche and V. Vianu, eds., pp. 316-330, Jan. 2001.
[4] P. Buneman and W.-C. Tan, “Provenance in Databases,” Proc. ACM SIGMOD, pp. 1171-1173, 2007.
[5] Y. Cui and J. Widom, “Lineage Tracing for General Data Warehouse Transformations,” The VLDB J., vol. 12, pp. 41-58, 2003.
[6] S. Czerwinski, R. Fromm, and T. Hodes, “Digital Music Distribu- tion and Audio Watermarking,” http://www.scientificcommons. org/43025658, 2007.

[7] 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.
[8] F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video," Signal Processing, vol. 66, no. 3, pp. 283-301, 1998.
[9] S. Jajodia, P. Samarati, M.L. Sapino, and V.S. Subrahmanian, "Flexible Support for Multiple Access Control Policies," ACM Trans. Database Systems, vol. 26, no. 2, pp. 214-260, 2001.
[10] 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