International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Identifying the Data Leaker by Using Sample Data Request Strategy

Lata Dudam[1], Dr. S. S. Apte[2]

[1]Walchand Institute of Technology, Solapur, India

[2]HOD, CSE Department, Walchand Institute of Technology, Solapur, India

Abstract: ***In course of doing business, sensitive data must be handed over to trusted third parties. The owner of the data is called Distributor and supposedly trusted third parties as the agent. Some of distributor's data may be leaked and found at unauthorized place. Our goal is to detect the distributor's sensitive data have been leaked by agents and identify the agent who leaked the data. We have used the fake objects which are realistic but fake records, are added to original data, which helps in identifying the guilty agents (the agent who leaked the data). We have used Sample data request allocation strategy for data distribution and we have implemented probability distribution system, to identify the guilty agents and leaked data.***

Keywords: Allocation strategies, data leakage, fake object, guilty agent.

1. Introduction

In real world, for business deals, some sensitive data has to be hand over to some trusted third parties. For ex, a hospital may give patient records to researchers for detecting the new treatments. And a company may have partnerships with some other companies that require sharing of sensitive data. The agents may leak the data. Our goal is to identify the leakage of data and guilty agents (the agents who leaked the data)[5].

Previously, watermarking techniques were used to reach the goal. But watermark technique was results in modification of original data. In this paper, we are using the realistic fake records which are added to original data, which helps in identifying the leakage.

We developed a model for identifying the guilty agents which we refer to as "Guilt model". We also implemented the algorithms for distributing data to agents. we also consider the option of adding "fake" records to the distributed data. Such objects don't 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 [5]. If an agent was given one or more fake records that were leaked, then the distributor can be more confident that agent was guilty.

1.1. Guilty Agent

The agents who have leaked the distributor's sensitive data is called the guilty agent. Suppose, distributor having data set S which is distributed to all agents. if the agent leak the set T €S, then the agent is called guilty.

1.2. Fake Objects

Fake objects are objects generated by the distributor that are not in set S. The objects are designed to look like real objects, and are distributed to agents together with the S objects, in order to increase the chances of detecting agents that leak data.

2. Literature Review

The detection of guilty agent approach is associated to the data provenance problem [11]: tracing the linearity of objects implies the detection of the guilty agents. Shouhuai Xu, Qun Ni and Elisa Bertino and Ravi Sandhu worked on provenance problem.

James Cheney, Laura Chiticariu and Wang-Chiew Tan[2] was developed a model on provenance problem for database system. The suitable prepared solutions are domain specific, such as lineage tracing for data warehouses [12].

R. Agrawal and J. Kiernan were developed "Watermarking Relational Databases,"Proc. 28th Int'l Conf. Very Large Data Bases (VLDB '02), VLDB Endowment, pp. 155-166 in 2002[6]. Watermarks were initially worked for images, video and audio data [9] whose digital representation includes some redundant data.

R. Sion, M. Atallah, and S. Prabhakar presented "Rights Protection for Relational Data," Proc. ACM SIGMOD, pp. 98-109, 2003[8].

S. Jajodia, P. Samarati, M.L. Sapino, and V.S. Subrahmanian[10] and B. Mungamuru and H. Garcia-Molina[1] developed database system by using access control policies and privacy techniques.

Panagiotis Papadimitriou, Hector Garcia-Molina [5] and Rudragouda G Patil[7] developed a model for data leakage detection using allocation strategies in 2011.

3. Implementation Methods

In the original data, by adding fake records, the data is distributed to number of agents. After distribution of data, we find the probability by considering the threshold 0.50. i.e. if probability of agents who leaked data is greater than 50% then we consider it as guilty agent[7].

Licensed Under Creative Commons Attribution CC BY

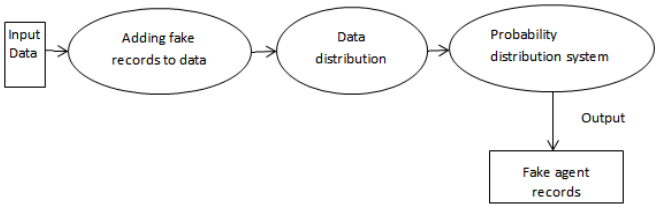


Figure 1: Overall system

3.1. Sample data request

With sample data requests [5][7] agents are got the particular objects from distributor. Hence, object sharing is not explicitly defined by their requests. The distributor distributes certain objects to multiple agents by its own decision. The more objects are shared among different agents; the more difficult it is to detect a guilty agent.

In this algorithm, we select some agents and distribute original data with fake records to selected agents. And by calculating the probability of agent's data, the distributor find the leaked data with leaker.

3.2. Guilt Agent Model

In this model, probability (guilty agents) is calculated as:

P (agent leaked data) = leaked data/number of agents.

If multiple agents having same data and distributor the same data leaked by multiple agents then distributor find the sum objective to overcome the difficulty of overlapping of agents.

Algorithm . Allocation for Sample Data Requests ($\overline{SF}$)
Input: $m_1, \ldots, m_n, |T|$ ▷ Assuming $m_i \leq |T|$
Output: $R_1, \ldots, R_n$
1: $a \leftarrow \mathbf{0}_{|T|}$ ▷ $a[k]$:number of agents who have received object t_k
2: $R_1 \leftarrow \emptyset, \ldots, R_n \leftarrow \emptyset$
3: $remaining \leftarrow \sum_{i=1}^{n} m_i$
4: **while** $remaining > 0$ **do**
5: **for all** $i = 1, \ldots, n : |R_i| < m_i$ **do**
6: $k \leftarrow$ SELECTOBJECT(i, R_i) ▷ May also use additional parameters
7: $R_i \leftarrow R_i \cup \{t_k\}$
8: $a[k] \leftarrow a[k] + 1$
9: $remaining \leftarrow remaining - 1$

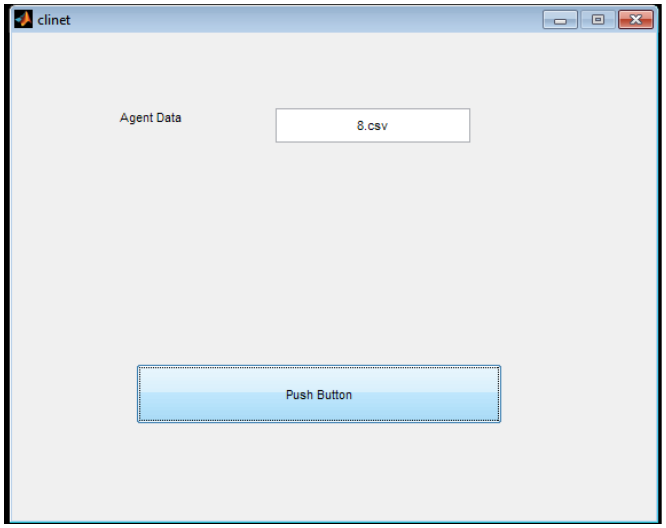


Figure 2: Client window

4. Experiment and Result

Our goal is to identify the guilty agents with leaked data. This paper is implemented client –server architecture. We see execution steps of project as follows:

4.1. Data Distribution :

Distributor distributes original data (customer's email ids). To number of agents by adding some fake records (fake email ids) to it. Figure 2 shows distribution process.

4.2. Distributor starts detection:

After successful completion of data distribution, distributor starts the server process to identify who is the leaker.

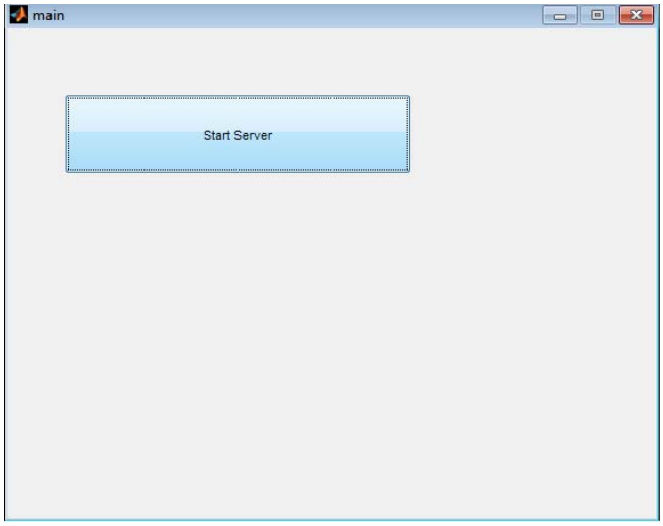


Figure 3: Server window

4.3. Detection of data leakage:

Once we start the server process, we move to one new window, which shows data leakage by agents. The result shows list of new mails received to all fake mails. From these fake mails, we can say that some of agents send the distributor's data to some other third parties.

Licensed Under Creative Commons Attribution CC BY

As we know the fake mail ids are created by distributor which helps in identifying the leakage and guilty agents.

This shows that, once distributor sends the data, agent leak it to some third party who is in partnership with guilty agent, but as it is a fake record created by distributor. So we can get easily the guilty agent.

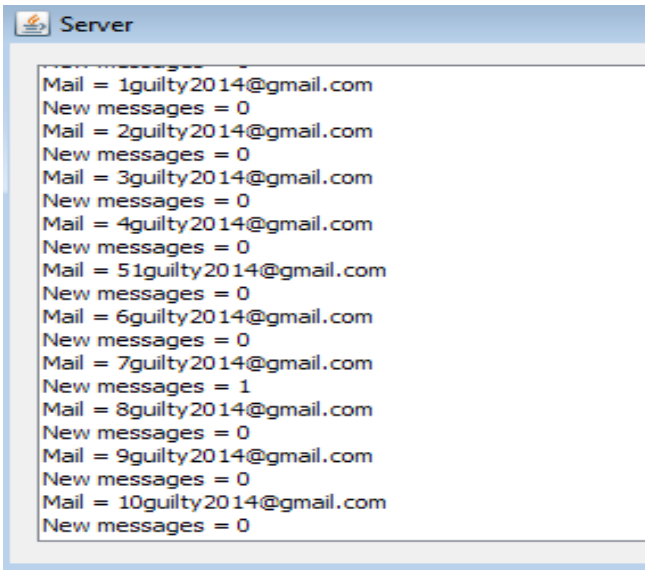


Figure 4: detection of leakage

4.4. Probability calculation:

To calculate the guilt probabilities and differences, we use throughout this section p ¼ 0:5. Although not reported here, we experimented with other p values and observed that the relative performance of our algorithms and our main conclusions do not change. If p approaches to 0, it becomes easier to find guilty agents and algorithm performance converges. On the other hand, if p approaches 1, the relative differences among algorithms grow since more evidence is needed to find an agent guilty.
We consider some threshold i.e. probability at 50%, and check the agents probability by applying the condition that

If (p (agent)> 50%) then

The agent is guilty.

The following window shows a4 agent having probability 61% which is greater than 50%.



Figure 5: Probability calculation

4.5. Guilty agent:

This widow shows the guilty agent.

Finally, we calculated the probability by giving each agent a point if he forward mail to some other agent and if that other agent is fake record, we immediately get known that agent is guilty.

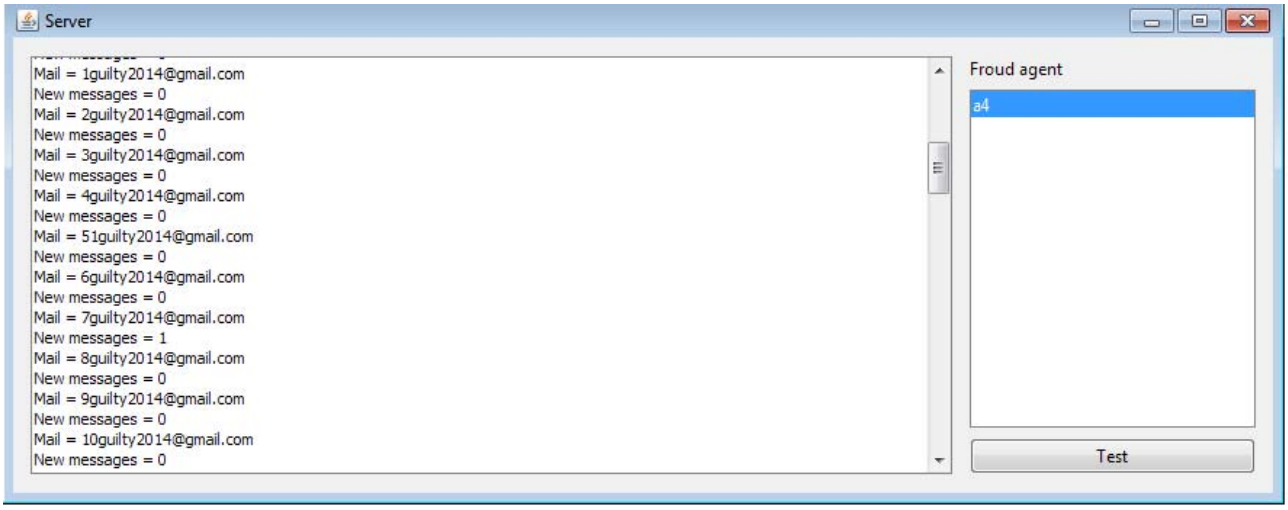


Figure 6: Guilty agent list

Licensed Under Creative Commons Attribution CC BY

5. Conclusion

In real world, there is no need to hand over the data to third parties. And even if we distribute it, we must take care by encrypting or watermarking the data so that it should not be leaked. In many cases, the agents with whom we are working that may not be completely trusted and we do not get who leak the data from given agents. To identify this previously watermarking technique were used, using watermarks original data modified. Therefore, we implemented the sample data request allocation strategy by adding some fake records to our sensitive data which results in detecting and identifying the guilty agents.

6. Future Scope

We can apply this method for business application and hospital management.

References

[1] B. Mungamuru and H. Garcia-Molina, "Privacy, Preservation and Performance: The 3 P's of Distributed Data Management," technical report, Stanford Univ., 2008.
[2] James Cheney, Laura Chiticariu and Wang-Chiew Tan. Provenance In Data Base, Foundations and TrendsR in Databases Vol. 1, No. 4 (2007) 379–474-c 2009.
[3] James Michaelis, Jim McCusker, Deborah L. McGuinness, Linked provenance data: A semantic Webbased approach to interoperable workflow traces, Received in revised form 12 October 2010.
[4] 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.
[5] Panagiotis Papadimitriou, Hector Garcia-Molina, "Data Leakage Detection" , IEEE transactions on knowledge and data engineering, vol. 23, no. 1, January 2011.
[6] 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.
[7] Rudragouda G Patil, "Development of Data leakage Detection Using Data Allocation Strategies International Journal of Computer Applications in Engineering Sciences [VOL I, ISSUE II, JUNE 2011, [ISSN: 2231-4946].
[8] R. Sion, M. Atallah, and S. Prabhakar, "Rights Protection for Relational Data," Proc. ACM SIGMOD, pp. 98-109, 2003.
[9] S. Czerwinski, R. Fromm, and T. Hodes. Digital music distribution and audio watermarking.
[10] 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
[11] Shouhuai Xu, Qun Ni and Elisa Bertino and Ravi Sandhu. A Characterization of The Problem of Secure Provenance Management, University of Texas at San Antonio, ISI 2009, June 8-11, 2009.
[12] Y. Cui and J. Widom. Lineage tracing for general data warehouse transformations. In The VLDB Journal, pages 471–480, 2001.

Licensed Under Creative Commons Attribution CC BY