International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 8 (2014), pp. 817-822

http://www. irphouse.com

Guilty Agent Detection Model

Shrey Mahajan[1], Parth Joshi[2] and Roheen Chaturvedi[3]

[1, 2 and 3]Department of Computer Engineering, D.Y.Patil College of Engineering Akurdi, Pune, India

ABSTRACT

A data proprietor has provided the handlers with the critical data to carry out their respective objective. But some of the provided data is found on unwanted place like a website. So the proprietor must consider that the agent has intentionally leaked the data. So we propose the model that improves the chances of identifying the guilty agents. This model rely on the technique which involves the addition of the fake data along with the real data but not on the perturbation of data (e.g. watermarking).The insertion of real looking fake data improves the chances of detecting the person behind the leakage without doctoring the original data.

Keywords- Guilty Agents, Data Proprietor, Data Leakage, Fake Objects, Leakage Detection.

1. INTRODUCTION

Handler in the multiagent system represents the different partners who have their own specific and sometimes contradictory interest and intentions. They would try to achieve their own target even at the expense of others.

Recent studies have shown a drastic increase in the number of handlers handling the data. These handlers, who represent the organisation or the person in the business have the essential knowledge of the data and can participate in the important decision making meetings. But these agents can try to put their own benefit above the organisation.

While doing business many times sensitive data must be handed over to supposedly handler i.e. the third party agent. This increases the risk of secured information falling into unauthorized hands. For example an MNC's like Walmart may give their customers records to the handlers/agency for research to build up new marketing strategies. Similarly, an agency may have partnership with the other MNC's that requires the same kind of sharing. If the handler is not trustworthy than the one MNC can outsource the data of another. The proprietor of the data is labelled

as the distributor and the supposedly third parties the agents. In this project, our objective is to detect what data has been leaked on the internet (website) and which agent is responsible for the leakage.

Techniques like Perturbation is very useful in such situations. Perturbation is the technique where the data is made less sensitive by modifying the data before handing it over to the agents. But in some cases the agent may need the original data to carry out the research.

2. OBJECTIVE

Data leakage can be defined as the release of secure/private information in an untrusted environment. The objective is to identify how much data has been leaked and which agent is responsible behind the leakage of the data assuming that the leakage is because of the agents. The data allocation strategies help the proprietor to distribute the data among the agents wisely. Fake tuples are going to be added among the original data to identify the guilty, to address this problem four instances are specified. And depending on the type of data request made fake tuples are added to the original data without changing the original data.

3. EXISTING SYSTEM

Generally, leakage detection is handled by the watermarking technique i.e. a unique code is embedded in every copy of the data and later if that copy is found in the hands of unapproved third party, the leaker can be identified. Though watermarks are useful in some cases but they also requires the modification of the original data. And watermarks can easily be removed or destroyed with the help of different software or techniques.

4. PROPOSED SYSTEM

It is possible to predict that the agent is responsible behind the leak, based on the overlap of the data provided to him with the leaked data found and the data given to the other agents. The presented algorithm implement different data distribution strategies that increases the chances of proprietor to detect the person behind the leakage. It is shown that if the data is distributed wisely than one can easily identify the guilty agent.

In this project the model to detect the guilty agent is created. The fake objects are added to data that is to be distributed. Such object don't affect the original data but appear as much real to the agents. So one can say that these fake objects act as the watermark for the proprietor without changing the original data. And if the data is leaked on the internet (displayed on some URL's) one can identify the guilty agent who has leaked the data with the help of identifying the one or more fake object leaked along with the original data.

5. PROBLEM DEFINITION

The proprietor's data allocation to the agent has one limitation and only one aim. The proprietor's limitation is to satisfy the request of the agents by providing them with the particular data they requested or providing them with all the data that satisfies their conditions. And the aim is to detect the agent who leaks the provided data.

The limitation is considered as strict. The distributor cannot deny the request of the agents and he cannot give agents the different version of same data. So for this, fake tuple distribution is the only way to get relaxation from this limitation. It makes the detection objective ideal. It maximize the chances of detecting the guilty agents.

5.1 Problem Setup and Notation

A proprietor owns a set $D=d_1, d_2 \ldots d_n$. And he wants share some of the data with the set of the agents $a=a_1, a_2 \ldots\ldots a_n$, and doesn't want objects to be leaked. Any agent a_i is going to receive the subset of objects depending on the type of request i.e. either sample request or the explicit request.

Sample Request SR= SAMPLE (D; mi): Any subset mi records from D can be given to the agent. Explicit Request ER= EXPLICIT (D; Cond): Agent receives all the D objects that meet his conditions. Our model can easily be extended to meet the request of the agent and satisfy their conditions. And we are not concerned with the randomness of a sample. If the sample request is made then the data objects can be given randomly.

5.2 Related work

The guilt decision approach which we presented over here is related to the data provenance problem: tracing the lineage of an S object leads to the detection of the guilty agents. Suggested solutions are domain specific whereas our formulation with the objects and sets is simple as we don't modify the object sets i.e. data transformation from Ri sets to S. As far as allocation of data is concerned we are adding the fake objects which act as the watermarking techniques. Our approach and the watermarking is same in the sense of providing the agent some additional information which can later be tracked. However watermark modifies the original data whereas our approach doesn't make any change in the original data

6. MODULE DESCRIPTION

6.1 Modules

- Proprietor Module

  Data Allocation

  Fake Objects

- Detection of Guilty Agent
- Agent Module

6.2 Proprietor Module

It consist of the admin application which is going to manage the details of the agents.

It provides the options like addition, deletion of the agents. The whole database is maintained on Apache Tomcat or Glass Fish server. This server is not only responsible for honouring the data request of the agents (both sample as well as explicit) but also for the addition of the generated fake tuples. The database stored on the server is going to keep the record of all the requests made and the requests which are honoured. It is also going to maintain the record of the fake tuples supplied to various agents making it easy to identify the guilty agents.

6.2.1 Data Allocation

We are going to handle two type of requests: sample and explicit. Fake tuples are going to be generated by the proprietor of the data. These fake tuples are the data objects which are not in set D. These fake tuples are made to look like the original and only the distributor can differentiate between the original and the fake tuples. So these fake tuples are distributed to the agent along with the original data which help us in identifying the agent who has leaked the data. There are four different instances depending upon the type of request and whether fake objects are allowed or not.

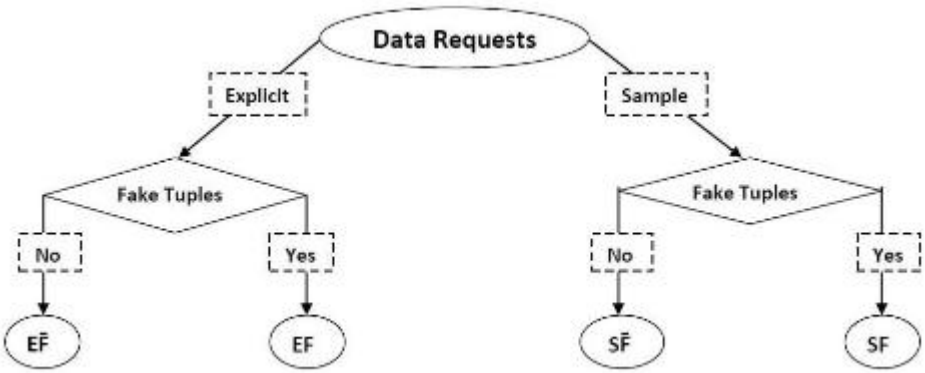

Figure.1 *Leakage Problem Instances*

We represent our problem instances with names EF, EF, SF, and SF where E stands for explicit and S stands for sample. F for the fake objects and F for fake objects not allowed. For the ease of our model we are assuming that agents can make an explicit or sample request at one instance.

6.2.2 Fake objects

The proprietor is going to add the fake object (which is similar to the real object) between the real entities in order to detect the guilty agent. While handling the same data request of two agents different fake tuples are going to be generated and it is assumed that these agents don't disclose their data to each other. Adding of these fake tuples causes less problems then perturbing the real object. For example distributed data be the medical record of the patient and the agents be the pharmaceutical company. In this case the modification in the actual patient records can be undesirable. But adding the extra fake patient record will be harmless. A file is

maintained to remember the fake data give to the agents which help in identifying the guilty agent.

6.3 Detection of Guilty Agent

Suppose after providing the agent with the objects the proprietor discovers that the data has been leaked and is displayed on the website. Since agents have that data it is reasonable to suspect that some agent has leaked the data. However the agent can argue over their innocence they may say that the data is leaked by someone else. For example if the two agent have the same data they can blame one another. At this moment the fake data comes in use in identifying the guilty agent. So this module is going to help in detecting the guilty agent looking for the fake data among the leaked data to identify the guilty agent. It consist of the sub modules like Web Data Extractor and Search module which will analyse the predefined URL's for the leaked data and search for the data automatically. Then the HTML tree is going to be generated. Once the tree is generated the data is going to be downloaded and the TAG tree evaluation is going to take place in which in which the fake tuples are going to be arranged at the leaf nodes of the tree. Then with the help of string matching fake tuples can be compared with the data supplied to the agents. After string matching it is easy to identify the leakage and the leaked data can help in identifying the agent behind the leakage.

6.4 Agent Module

It consist of different agents which are provided with the option to make a request for the data. Once their request is met they are given chance to leak the data on the internet. The chance of leaking the data will be provided with the help of FTP uploader. In which the agent has to specify the URL of the website on which he wants to leak the data. The agent who has leaked the data is going to be called as the guilty agent and the other as the innocent or non-malicious agents.

7. SYSTEM IMPLEMENTATION

The proprietor is going to manage the whole database including the details of all the handler/agents. All the agents can make the request for the data only when they have registered with the proprietor. The agents have to fill all the registration details and the distributor is going to validate the registration of the agent. Once the agent is registered he can request for the data accordingly his request is met. Only distributor can supply the data to the agent by adding the fake objects along with the original data. And distributor is maintain the record file of the data supplied to the handler along with the fake objects.

8. CONCLUSION

In today's world, with the different technologies evolving every minute there is no surety that our data is safe for the sharing. At this moment handlers come in place. MNC's trust these handlers with the data to carry out the research or the asked

objective. But not every agent is trustworthy, some are malicious also. So to protect the data from these malicious agents this model can be used. This model improves the chances of detecting the leakage and the person behind the leakage. It has been shown that distributing the data objects wisely can improve the chances of identifying the leak. And the insertion of fake objects instead of perturbation is helpful as the original data is not modified.

9. REFERENCES

[1] P. Papadimitriou, Student Member, IEEE, and Hector Garcia-Molina, Member (2011), Data Leakage Detection, *IEEE transactions on knowledge and data engineering*, vol. 23, no. 1, pp. (2-6).

[2] S.Umamaheswari and H.A.Geetha (2011) Detection of Guilty Agents, *Proceedings of the National Conference on Innovations in Emerging Technology*.

[3] S.A. Kale, Prof. S. V. Kulkarni (2012), Data Leakage Detection: A Survey, *IOSR Journal of Computer Engineering (IOSRJCE)* ISSN: 2278-0661 Vol. 1, Issue 6, pp. 32-35.

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

[5] P. Bonatti, S.D.C. di Vimercati, and P. Samara (2002), An Algebra for Composing Access Control Policies, *ACM Trans. Information and System Security*, vol. 5, no. 1.

[6] Y. Cui and J. Widom (2003), Lineage Tracing for General Data Warehouse Transformations, *The VLDB J.*, vol. 12, pp. 41-58.

[7] Y. Li, V. Swarup, and S. Jajodia (2005), Fingerprinting Relational Databases: Schemes and Specialties, *IEEE Trans. Dependable and Secure Computing*, vol. 2, no. 1.