Discovery of Data Leakage Detection

Vaishnavi Patil[1], Rupali Pangare[2], Mayuri Pisal[3], Prof. Shivsagar Gondil[4]
Department of Computer Engineering
[1,2,3,4] Bharati Vidyapeeth College Of Engineering, Navi Mumbai

Abstract- *A data distributor has provided confidential data to a series of supposedly reliable agents (third parties). Some data is filtered and found in an unauthorized location (for example, on someone's web or laptop). The distributor must assess the probability that the leaked data will come from one or more agents, rather than being collected independently by other means. Data allocation strategies (via agents) have been proposed that improve the likelihood of identifying losses. These methods are not based on alterations of published data (eg watermarks). In some cases, the distributor can also inject "realistic but false" data to further improve our chances of identifying losses and identifying the guilty party.*

I. INTRODUCTION

In the course of business, sometimes confidential data must be delivered to trusted third parties. For example, a hospital can provide patient documentation to researchers who will design new treatments. Likewise, a company may have a partnership with other companies that need to share customer data. Another company can outsource its data processing, so the data must be delivered to other companies. We call the data owner the distributor and the third parties, presumably trusted, the agents. Our goal is to detect when agents have disclosed confidential distributor data and, if possible, identify the leaked data agent. Consider the applications where the original confidential data cannot be disturbed. The disturbance is a very useful technique in which the data are modified and made "less sensitive" before being delivered to the agents. For example, you can add random noise to certain attributes, or you can replace exact values based on intervals. However, in some cases it is important not to change the data of the original distributor. For example, if a subcontractor is doing our payroll, he must have the exact salary and customer identification numbers. If medical researchers treat patients (instead of simply calculating statistics), they may need accurate patient data. Traditionally, leak detection is managed by watermarks, for example, a unique code is embedded in every distributed copy. If an unauthorized party discovers this copy, the escape agent can be identified. Watermarks can be very useful in some cases, but, again, imply some changes in the original data. Furthermore, watermarks can sometimes be destroyed if the recipient of the data is malicious. In this article we study discrete techniques to detect the losses of a setof objects or records. (For example, data can be found on a website or can be obtained through a legal discovery process). At this point, the distributor can assess the probability that the leaked data will come from one or more agents, rather than being collected independently by other means. Using an analogy with cookies stolen from a jar of cookies, if we have caught Freddie with a single cookie, you can object that a friend has given you the cookie. But if we take Freddie with 5 cookies, it will be much harder for him to say that his hands were not in the cookie jar. If the distributor sees "sufficient evidence" that an agent has leaked data, he may stop doing business with him or initiate a legal proceeding.

1.1 Problem

Suppose that a distributor has a set T = {t1, tm} of value data objects. The dealer wants to share some of the objects with a series of agents U1, U2 ... A, but you want the objects to leak for another third. An agent Ui receives a subset of Ri objects belonging to T, determined by an example of request or explicit request, sample application Ri = SAMPLE (T, mi): Any subset of records from my T can be given Ui. Re = EXPLICIT explicit request (T, Condi): the agent Ui receives all the T objects that satisfy the Condi. T objects can be of any type and size, for example, it could be tuples in a relationship or relationships in a database. After giving objects to the agents, the distributor discovers that a series S of T has been filtered. This means that a third party called objective has been found in possession of S. For example, this goal may display S on your site, or perhaps as part of a legal discovery process, the target

S delivered to the distributor. From the U1, U2 ... Un, the agents have some of the data, it is reasonable to suspect that they are data loss. However, agents can claim that they are innocent and that S data was obtained from the target by other means.

1.2 Purpose of the project

Data loss detection techniques are designed so that users can monitor if data has been lost and trace sources of data loss. Many times we find cases in which the filtered data are in unauthorized places. For example, we may find confidential data stored on a laptop or unauthorized website. At that time, it becomes important to trace the source of data loss. To this end, we propose an improved data loss detection technique to track unauthorized leakage sources through the use of a data allocation strategy through various agents. The strategy allows the user to transfer data to users by considering the receivers as agents to whom the data are assigned along with some undetectable alterations based on id. These alterations allow our system to track the source of leaked data as soon as it is in an unauthorized source. In this system, we aim to identify data loss by storing data based on agents. Our system is designed to detect data in formats (.txt,.jpg and .bmp).

1.3 Scope of the project

Our future work includes

- The investigation of the guilt models of agents who capture loss scenarios that have not been studied in this document. For example, what is the appropriate model for cases where agents can collude and identify false tuples?

- The extension of our allocation strategies so that they can manage the requests for agents online (the strategies presented assume that there is a fixed group of agents with requests known in advance.

II. REVIEW OF THE LITERATURE

2.1 General Introduction

- The guilt-detection approach we present is related to the data source problem, to trace the lineage of S objects essentially involves the detection of guilty agents.
- Solutions Suggested solutions are domain-specific, such as lineage tracking for data warehouses, and involve preliminary knowledge of how a data view is created from data sources.
- Fill Watermarks were initially used for images, video and audio data whose digital representation includes considerable redundancy. The watermark is similar in the sense that it provides agents with some kind of information that identifies the recipient. However, by its very nature, a watermark changes the element that is marked. If the object to be tagged with a watermark cannot be edited, a watermark cannot be inserted. In such cases, the methods that attach watermarks to the distributed data are not applicable.
- Recently, the works have also studied the inclusion of brands in relational data.
- There are also many other work on mechanisms that allow only authorized users to access confidential data through access control policies. Such approaches prevent, in a sense, data leakage by sharing information only with reliable parts. However, these policies are restrictive and can make it impossible to meet agents' requests.

III. EXISTING SYSTEM

3.1 Disturbance

The application in which confidential confidential data cannot be disturbed was considered. The disturbance is a very useful technique in which the data are modified and made "less sensitive" before being delivered to the agents. For example, you can add random noise to certain attributes, or you can replace exact values based on intervals. However, in some cases it is important not to change the data of the original distributor. For example, if an external supplier is doing our payroll, you must have the exact salary and bank account number of the customer. If medical researchers treat patients (instead of simply calculating statistics), they may need accurate patient data.

3.2 Watermark

Traditionally, leak detection is managed by watermarks, for example, a unique code is embedded in every distributed copy. If this copy is later discovered by an unauthorized party, the escape agent can be identified. Watermarks can be very useful in some cases, but again imply some changes to the original data. Furthermore, watermarks can sometimes be destroyed if the recipient of the data is malicious.

Disadvantages

- Consider applications where sensitive confidential data cannot be disturbed. The disturbance is a very useful technique in which the data are modified and made "less sensitive" before being delivered to the agents.
- However, in some cases it is important not to alter the data of the original distributor.
- Traditionally, leak detection is managed by watermarks, for example, a unique code is embedded in every distributed copy.
- If this copy is later discovered in the hands of an unauthorized party, the escape agent can be identified.
- Watermarks can be very useful in some cases, but, again, involve some changes to the original data.
- In addition, watermarks can sometimes be destroyed if the recipient of the data is malicious.

IV. PROPOSED SYSTEM

Discrete techniques have been studied to detect losses from a set of objects or registers. After giving a set of objects to the agents, the distributor discovers some of those same objects in an unauthorized location. (For example, data can be found on a website or can be obtained through a legal discovery process). At this point, the distributor can assess the probability that the leaked data will come from one or more agents, rather than being collected independently by other means. Using an analogy with cookies stolen from a jar of cookies, if Freddie with only one cookie has been cached, you can object that a friend has given you the cookie. But if Freddie with 5 cookies has been cached, it will be much harder for him to argue that his hands were not in the cookie jar. If the distributor sees "sufficient evidence" that an agent leaked data, he may stop doing business with him or start a legal proceeding.

A model was developed to evaluate the "fault" of the agents. An algorithm has been proposed to distribute objects to agents, so as to improve our chances of identifying a leak. The option to add "fake" objects to the distributed set was also considered. These objects do not correspond to real entities, but they seem realistic for the agents. In a sense, fake objects act as a type of watermark for the entire set, without modifying any single member. If it turns out that an agent has been given one or more fake items that have been leaked, then the dealer may be more confident that the agent was at fault.

Benefit

- After giving a set of objects to the agents, the distributor discovers some of those same objects in an unauthorized location.
- At this point, the distributor can evaluate the probability that the leaked data will come from one or more agents, rather than being collected independently by other means.
- If the distributor sees "sufficient evidence" that an agent leaked data, he may stop doing business with him or start a legal proceeding.
- Develop a model to evaluate the "fault" of the agents.
- We also present algorithms to distribute objects to agents, in order to improve our chances of identifying a filter.

- Consider the option of adding "fake" objects to the distributed set. These objects do not correspond to real entities but appear.
- If it turns out that an agent has been given one or more fake items that have been leaked, then the dealer may be more confident that the agent was at fault.

V. SYSTEM MODULES

5.1 Data Assignment:

This module is designed primarily to transfer data from the distributor to the agents. The same form can also be used for the illegal transfer of data from agents authorized to other agents. The distributor intelligently distributes data to agents to improve the chances of identifying a guilty agent. Four instances of this problem can be addressed, depending on the type of data requests made by the agents and whether "fake objects" are allowed. The two types of requests managed are: sample and explicit. False objects are objects generated by the distributor. Objects are designed to look like real objects and are distributed among agents to increase the chances of detecting agents that filter data. All agents make explicit requests and all agents make sample requests. Results can be extended to handle mixed cases, with some explicit requests and some examples.

5.2 Guilt Model

This module is designed using the agent error model. When the distributor sends data to the agent, during the execution time, this module assigns a single false object in each tuple supplied to the agents. A copy of the data, which is transferred to the agents, is stored in the distributor's database. The distributor adds fake items to the distributed data to improve the effectiveness of detecting guilty agents. However, fake items can affect the correctness of what agents do, so they cannot always be allowed. In this module, the distributor object set was disturbed by adding fake elements. In some applications, fake objects can cause fewer problems that disturb real objects. For example, let's say that distributed data objects are patient records and agents are researchers. In this case, even minor changes to actual patient record records may be undesirable. However, adding some fake items can be acceptable.

5.3 Agent Guilt Model:

This module is designed primarily for the determination of false agents. This module uses fake objects (which are stored in the error model module database) and determines the error agent along with the probability. Once the distributor finds your data in unauthorized locations, you can compare the release data with your copy of the data, which is distributed to the agents, and then you can find the fault agent. To calculate this probability, we need an estimate of the probability that values can be "guessed" by the target. We use the probability of guessing to identify the agents that have leaked information. The probabilities are estimated based on experiments.

5.4 Find Probability

When the distributor finds data leaked to unauthorized locations such as websites or laptops, the distributor must be able to find out who the guilty agent is. For this, you can find the probability of the amount of data allocated for a particular agent by the number of records found in unauthorized locations.

From this it is clearly concluded that the agent is more likely to be the failure agent. False objects are used here to confirm the faulty agent more clearly.

VI. CONCLUSION

In our data leak detection project, we presented a robust cryptographic technique for relational data that incorporates cryptographic bits in data statistics. The cryptographic problem has been formulated as a limited optimization problem that maximizes or minimizes an occultation function based on the bit to be embedded. The GA and PS techniques were used to solve the proposed optimization problem and to manage the constraints. In addition, we present a data partition technique that does not depend on special marker tuples to locate partitions and demonstrate its resistance to Cryptography synchronization errors. We have developed an efficient technique based on the threshold for cryptographic detection based on an optimal threshold that minimizes the probability of decoding error.

The strength of cryptography has been improved by the repeated incorporation of cryptography and by the use of the majority voting technique in the cryptographic decoding phase. On the other hand, the ability to recover cryptography has been improved by using multiple attributes. A proof of the implementation concept of our cryptography technique was used to conduct experiments using synthetic and real data. A comparison of our cryptography technique with the techniques illustrated above shows the superiority of our technique in elimination, alteration and insertion attacks.

REFERENCES

[1] Data Leakage Detection, an IEEE paper by Panagiotis Papadimitriou, Member, IEEE, Hector Garcia-Molina, Member, IEEE NOV-2010.

[2] Watermarking relational databases. In VLDB ’02: Proceedings of the 28th international conference on Very Large Data Bases, By R. Agrawal and J. Kiernan, pages 155–166. VLDB Endowment, 2002.

[3] An algebra for composing access control policies, By P. Bonatti, S. D. C. di Vimercati and P. Samarati, ACM Trans. Inf. Syst. Secur., 5(1):1–35, 2002.

[4] P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. In J. V. den Bussche and V. Vianu, editors, Database Theory - ICDT 2001, 8th International Conference, London, UK, January 4-6, 2001.