Information Leakage Detection And Prevention Using K-K-Anonymity Algorithm

P.Parthiban[1], Dr V.Saravanan[2]

[1]Dept of Information Technology

[2]HOD & Professor, Dept of Information Technology

[1, 2] Hindusthan college of Arts and science, Coimbatore

Abstract- *In an organization the information in sub-office is transferred to main office for that a data distributor is needed to given sensitive data to a set of supposedly trusted agents such as third parties. Some of the data is leaked and found in an unauthorized place for example on the web or somebody's laptop. The distributor is responsible for that the leaked data came from one or more agents, as opposed to having been independently gathered by other means.*

In this project the data allocation strategies across the agents that improve the probability of identifying leakages. These methods do not rely on alterations of the released data. In some cases we can also inject "realistic but fake" data records to further improve our chances of detecting leakage and identifying the guilty party. If any of the intruders tries to download our information will get fake data only and will be locked and others are not able to view the data.

I. INTRODUCTION

1.1 PROBLEM DEFINITION

The aim of the paper is to transfer the data in a secure manner. This project will eliminate the situation that the data is viewed by third person (i.e., hacking of data). In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties. Traditionally, leakage detection is handled by watermarking. In this project we use a technique called unobtrusive for detecting leakage of a set of objects or records.

II. EXISTING SYSTEM

Traditionally, leakage detection is handled by watermarking, e.g., a unique code is embedded in each distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious.

Similarly, a company may have partnerships with other companies that require sharing customer data. Another enterprise may outsource its data processing, so data must be given to various other companies. We call the owner of the data the distributor and the supposedly trusted third parties the agents.

III. PROPOSED SYSTEM

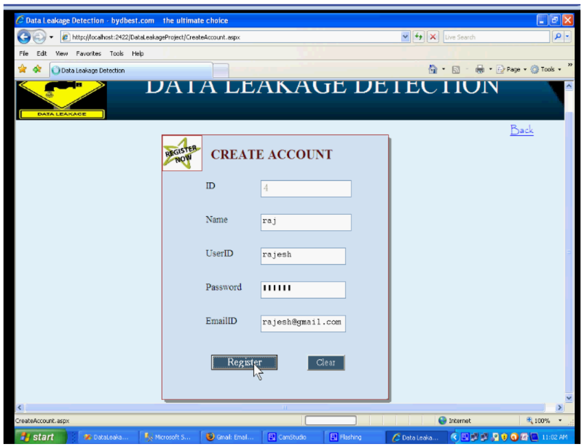
The proposed system should over come all the disadvantages of the existing system. The existing system is not functioning well due to some drawbacks. Thus the proposed system should minimize all the drawbacks. The goal is to detect when the distributor's sensitive data has been leaked by agents, and if possible to identify the agent that leaked the data.

In this section we develop a model for assessing the "guilt" of agents. This also presents algorithm for distributing objects to agents, in a way that improves the chances of identifying a leaker. Finally, it also considers 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 acts as a type of watermark for the entire set, without modifying any individual members. If it turns out an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty.

IV. DESCRIPTION OF MODELS

4.1 Data Allocation Module

The main focus of our project is the data allocation problem as how can the distributor "intelligently" give data to agents in order to improve the chances of detecting a guilty agent.


Data Leakage Detection - bydbest.com the ultimate choice
http://localhost:2422/DataLeakageProject/CreateAccount.aspx
File Edit View Favorites Tools Help
Data Leakage Detection
DATA LEAKAGE DETECTION
DATA LEAKAGE
Back
REGISTER NOW
Register
Clear
CreateAccount.aspx
Internet
100%
start

CREATE ACCOUNT

ID 4

Name raj

UserID rajesh

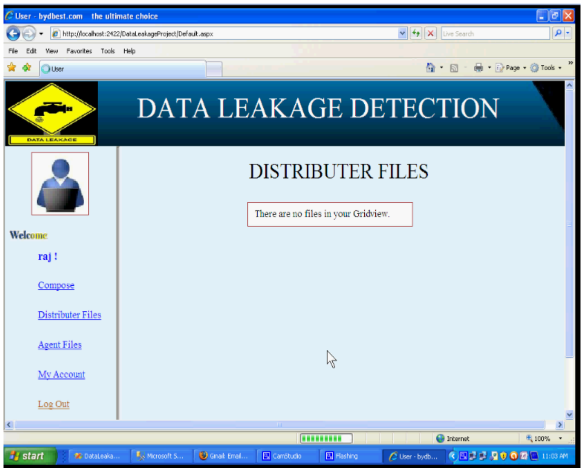
Password ||||||

EmailID rajesh@gmail.com

Figure 4.1.1 Registration form

4.2 Fake Object Module

Fake objects are objects generated by the distributor in order to increase the chances of detecting agents that leak data. The distributor may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. The use of fake objects is inspired by the use of “trace” records in mailing lists.


User - bydbest.com the ultimate choice
http://localhost:2422/DataLeakageProject/Default.aspx
File Edit View Favorites Tools Help
User
DATA LEAKAGE DETECTION
DATA LEAKAGE
Welcome
raj !
Compose
Distributer Files
Agent Files
My Account
Log Out
Internet
100%
start

DISTRIBUTER FILES

There are no files in your Gridview.

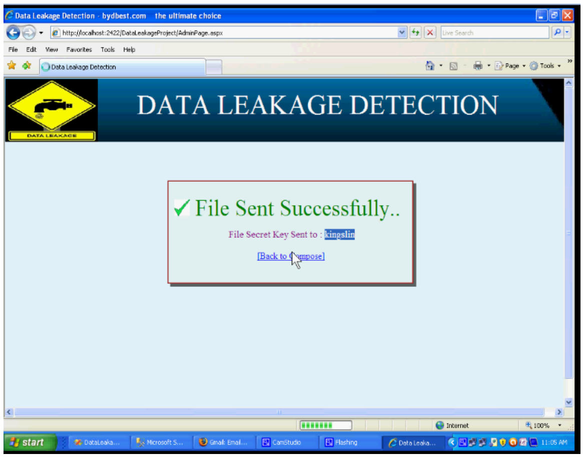
Figure 4.1.2 Distributed Files

4.3 Optimization Module

The Optimization Module is the distributor’s data allocation to agents has one constraint and one objective. The distributor’s constraint is to satisfy agent’s requests, by providing it with the number of objects it request or with all available objects that satisfy the conditions. The objective is to be able to detect an agent who leaks any portion of the data.

4.4 Data Distributor

A data distributor gives sensitive data to a set of supposedly trusted agents (third parties). Some of the data is leaked and found in an unauthorized place (e.g., on the web or somebody’s laptop).


Data Leakage Detection - bydbest.com the ultimate choice
http://localhost:2422/DataLeakageProject/AdminPage.aspx
File Edit View Favorites Tools Help
Data Leakage Detection
DATA LEAKAGE DETECTION
DATA LEAKAGE
[Back to Compose]
Internet
100%
start

✓ File Sent Successfully..

File Secret Key Sent to : kingslin

Figure 4.1.3 File Sent successfully

V. INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

5.1 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.
- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

VI. CONCLUSION

In a perfect world there would be no need to hand over sensitive data to agents that may unknowingly or maliciously leak it. And even if there is a need to hand over sensitive data, in a perfect world itcould watermark each object so that it could trace its origins with absolute certainty. However, in many cases it must indeed work with agents that may not be 100% trusted, and it may not be certain if a leaked object came from an agent or from some other source, since certain data cannot admit watermarks.

In spite of these difficulties, there have shown it is possible to assess the likelihood that an agent is responsible for a leak, based on the overlap of data with the leaked data and the data of other agents, and based on the probability that objects can be "guessed" by other means. Our model is relatively simple, but we believe it captures the essential trade-offs. The algorithms we have presented implement a variety of data distribution strategies that can improve the distributor's chances of identifying a leaker.

It have shown that distributing objects judiciously can make a significant difference in identifying guilty agents, especially in cases where there is large overlap in the data that agents must receive. Future work includes the investigation of agent guilt models that capture leakage scenarios that are not studied in this paper. For example, what is the appropriate model for cases where agents can collude and identify fake tuples? A preliminary discussion of such a model is available in 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).

VII. FUTURE ENHANCEMENTS

In future it is possible one to add new web pages without any problem with enhanced. As the technology used is a good one it is flexible for future enhancement and it is also possible to alter the front-end and back-end without any problem.

This web-based one is created effectively in a user-friendly manner and any new system that is developed in future must be incorporated or updated without any problem. So this will support enhancements in future.

REFERENCES

[1] H. Yao and B. Li, "An efficient approach for texture-based image retrieval", Neural Networks and Signal Processing, vol. 2, **(2003)**, pp. 1039-1043.

[2] H.-C. Lin, C.-Y. Chiu and S.-N. Yang, "Finding textures by textual descriptions, visual examples, and relevance feedbacks", Pattern Recognition Letters, vol. 24, no. 12, **(2003)** October, pp. 2255-2267.

[3] J. Zhang, G.-L. Li and S.-Wun, "Texture–Based Image Retrieval by Edge detection Matching GLCM", the 10th international Conference on High Performance computing and Communications, **(2008)**, pp. 782-786.

[4] P. Gangadhara Reddy, "Extraction of Image features for an Effective CBIR System", IEEE, **(2010)**, pp. 138-142.

[5] N. Chaturvedi, S. Agrawal and P. Kumar Johari, "A novel Approach of image retrieval based on texture", International Journal of Enhanced Research in Management & Computer Applications, ISSN: 2319-7471, vol. 3, no. 1, **(2014)** January, pp. 42-48.

[6] N. Puviarasan, Dr. R. Bhavani and A. Vasnthi, "Image Retrieval Using Combination of Texture and Shape Features", International Journal of Advanced Research in Computer and Communication Engineering, ISSN: 2319-5940, vol. 3, **(2014)** March, pp. 5873-5877.

[7] P. Howarth and S. Ruger, "Evaluation of Texture Features for Content Based Image Retrieval", Springer–Verlag Berlin Heidelbag LNCS 3115, **(2004)**, pp. 326-334.

[8] T. Deselaers, D. Keysers and H. Ney, "Feature for Image Retrieval: An Experimental Comparision", Springer -, **(2007)** November, pp. 1-22.

[9] R. N. Sutton and E. L. Hall, "Texture measurement for automatic classification of pulmonary disease", IEEE Trans. Compute., vol. C-21, **(1972)** July, pp. 667-676.

[10] K. R. M. Haralick, “Textural Features for Image Classification”, IEEE Transactions on Systems, Man and Cybernetics, vol. smc-3, no. 6, **(1973)** November, pp. 610-621.

[11] Hu, “Visual Pattern Recognition by Moment Invariants”, IRE Transactions on Information Theory, **(1962)**, pp. 179-187.

[12] D. Zhang and G. Lu, “A Geometric method to compute directionality features for texture images”, IEEE

[13] International Conference on Multimedia and Expo, ISSN: 978-1-4244-2571-6, **(2008)** April.

[14] N. Puviarasan, Dr. R. Bhavani and A. Vasnthi, “Image Retrieval Using Combination of Texture and Shape Features”, International Journal of Advanced Research in Computer and Communication Engineering, ISSN: 2319-5940, vol. 3, **(2014)** March, pp. 5873-5877.

[15] Brodatz texture database, www.ux.uis.no/~tranden/brodatz.html.

[16] http://www.agilemodeling.com/artifacts/systemflowdiagram.html

[17] http://www.vbdotnetheaven.com/

[18] http://www.sysimp.com

[19] http://en.wikipedia.org/wiki/winsock server

[20] http://www.testinggeek.com/testingtype.asp

[21] http://www.sei.cmu.edu./domain-engineering/usecasediagram.html

[22] http://en.wikipedia.org/wiki/windows-XP