Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 1, January 2014, pg.434 – 440

RESEARCH ARTICLE

Design & Implementation of Data Protection Server: Detect Guilty Agent & Protect Secure Data

Hema Donekar[1], Pratiksha Raut[2], Nita Janorkar[3], Shital Admane[4], Indu Mandwi[5]

Student, CSE, DBACER, Nagpur, India[1]
Student, CSE, DBACER, Nagpur, India[2]
Student, CSE, DBACER, Nagpur, India[3]
Student, CSE, DBACER, Nagpur, India[4]
Lecturer, CSE, DBACER, Nagpur, India[5]

ABSTRACT*: *This paper presents a proactive protect scheme based on Data protection Server. We propose an improved approach based on detection of leakage and identifying the guilty party.which enhances the security of data. A data distributor has given sensitive data to a set of supposedly trusted agents (third Parties). Some of the data are leaked and found in an unauthorized place (e.g., on the web or somebody's laptop). The distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other Means. We propose data allocation strategies (across the agents) that improve the probability of identifying leakages. These methods do not rely on alterations of the released data (e.g., watermarks). 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.

Keywords: Data Protection Server; Detect leakage of data; Find Guilty agent; Provide Security

I. INTRODUCTION

Data Protection server is one of the important factor in business dealings mainly in present trend. Companies share customer information with other companies who are in collaboration with that company. For this data security is important so data leakage detection will play important role. 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 our goal is to detect when the distributor's sensitive data have been leaked by agents, and if possible to identify the agent that leaked the data. The main objective of developing this data protection server is to trace the leaked data and misplaced having been independently assess the likelihood that the leaked data.A data distributor has given sensitive data to a set of data came from one a thereby. This server can trace those people which leak the confidential information to the unauthorized person. This server is used in banking, companies for their confidential transactions, collages, PM meetings etc. We propose data allocation strategies (across the agents) that improve the probability of identifying leakages. Supposedly trusted agents (third Parties). Some of the data are leaked and found in an unauthorized place (e.g. on the web or somebody's laptop).The distributor must or more agents, as opposed to Our aim is to provide data protection, provide protect the important data from unauthorized user and provide private key for transaction of data also detect the agent who leaked the distributor's sensitive data.

The remaining of this paper is organized as follows: section II provides proposed plan section III provides conclusion.

II. PROPOSED PLAN

This system will provide more amounts of hackers who are part of data leakage and take serious action against them. In this proposed system we use different methods to find out the hackers for example by generating fake objects, encryption of data. Algorithms are developed in the way that they are accessible to agents and by using these algorithms agent's information can be traced. We develop a model for assessing the "guilt" of agents. We also present algorithms for distributing objects to agents, in a way that it improves our chances of identifying a leaker.

A. *Guilty agent*

Suppose that after giving objects to agents, the distributor discovers that a set S T has leaked. This means that some third party, called the target, has been caught in possession of S. For example, this target may be displaying S on its website, or perhaps as part of a legal discovery process, the target turned over S to the distributor. Since the agents U1;...; UN have some of the data, it is reasonable to suspect them leaking the data. However, the agents can argue that they are innocent, and that the S data were obtained by the target through other means. For example, say that one of the objects in S represents a customer X. Perhaps X is also a customer of some other company, and that company provided the data to the target. Or perhaps X can be reconstructed from various publicly available sources on the web. Our goal is to estimate the likelihood that the leaked data came from the agents as opposed to other sources. Intuitively, the more data in S, the harder it is for the agents to argue they did not leak anything. Similarly, the "rarer" the objects, the harder it is to argue that the target obtained them through other means. Not only do we want to estimate the likelihood the agents leaked data, but we would also like to find out if one of them, in particular, was more likely to be the leaker.

1) Architecture

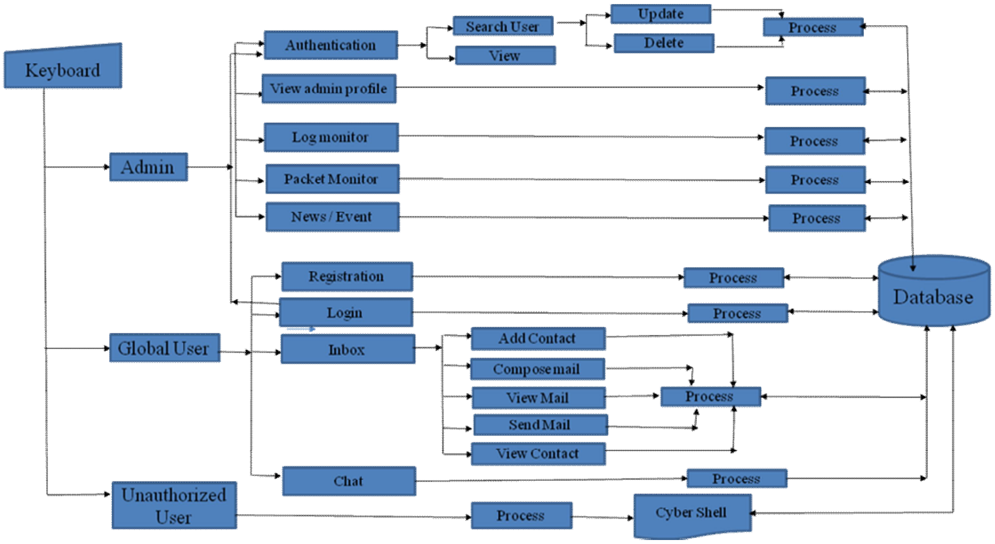


Fig:Architecture of data protection server.

The Architecture of the Data Protection Server is shown in above Figure. This figure shows essential components of the architecture: “Admin”, “Global User”, and “Unauthorised User”.

“Admin” is similar to server which handles the whole network and provides essential authority to the user.Admin is responsible for authenticate the user and handle news and events, update or delete user that he/she required.

“Global User” has to register them. And after authentication admin provides user name and password to the user using this password user can log in. Admin send tasks to the multiple users and provide key to each file using this key user can access that file.

“Unauthorized User” is like hacker that he has no authority to access confidential data which is sends by Admin to user.

2) Modules

- Admin:

Admin is behaving as Server which builds For the purpose of the security**.** Admin user handles the whole system, without the permission of the admin user any new user cannot handle the system. Administrative provide authority to new user after clarifying the all information Provided By user. Admin contain the confidential data to be secured are secured information towards the client .Admin can handle overall operation that should be performed during the transmission. Admin can create and manages also delete or modify the news and events record. Also admin can authenticate the client. Admin can view the contact enquiry of the site visitors, IP address of the site visitor. Admin can able to change his password also change his profile Picture.

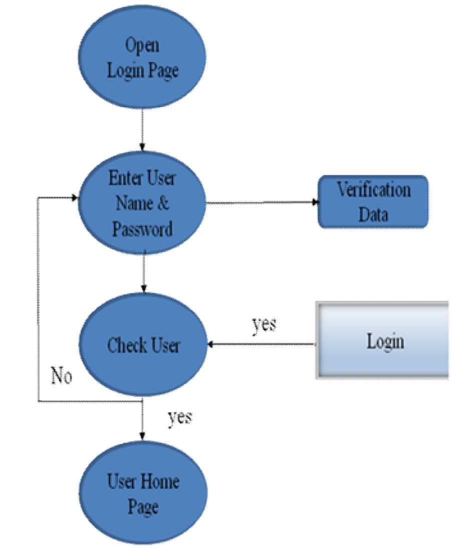


Fig: Data Flow Diagram For Login Form

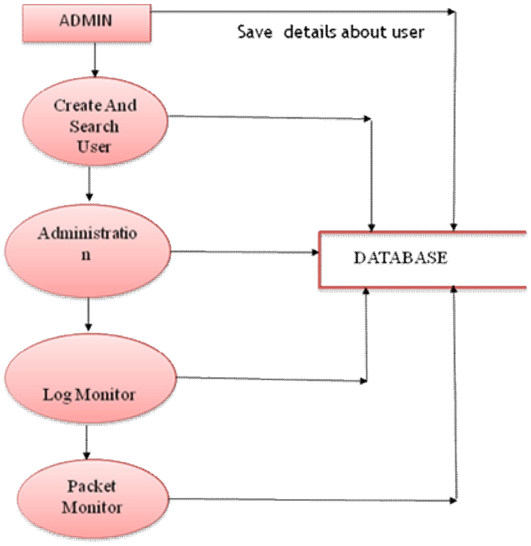


Fig : Data Flow Diagram For An Admin

- Client:

Global user is compulsorily taking a permission to Use the administrative system from Admin user. When admin sends the information to the user If the third party(unauthorized user) can access this confidential information that time admin check that whether the user grant the access to that third party or not. That the user leaks the information. First user has to register them. After the User login in inbox user can see some tasks which Is assign by Admin.

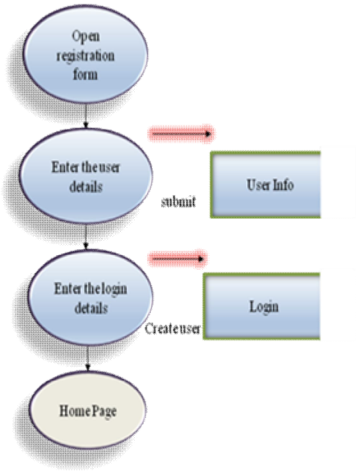


Fig: Data Flow Diagram For An Registration

2) Objective of Data Protection server

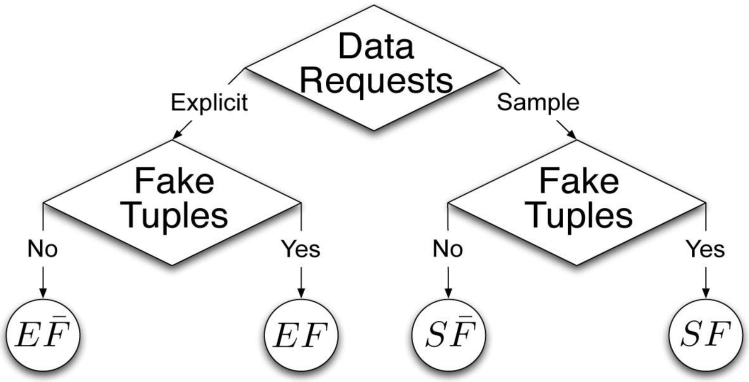
Data protection serve provides data protection and provide a private key for the transaction of the data..Data protection server protects the important data from unauthorized user. Data Protection Server provides the few layers for detect leakage of data at every stage. Data Protection Server is used to detect find the Agent who leaked the distributor’s sensitive data. This system will provide more amounts of hacker’s who are the part of data leakage and take serious action against them. This system will propose data allocation strategies that improve the probability of identifying the leakage.

B. *Fake Object*

The distributor may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. However, fake objects may impact the correctness of what agents do, so they may not always be allowable. The idea of perturbing data to detect leakage is not new e.g. [1]. However, in most cases, individual objects are perturbed e.g. by adding random noise to sensitive salaries, or adding a watermark to an image. In our case, we are perturbing the set of distributor objects by adding .In this case, even small modifications to the records of actual patients may be undesirable. However, the addition of some fake medical records may be acceptable, since no patient matches these records, and hence, no one will ever be treated based on fake records. Our use of fake objects is inspired by the use of “trace” records in mailing lists. In this case, company A sells to company B a mailing list to be used once (e.g. to send advertisements). Company A adds trace records that contain addresses owned by company A. Thus, each time company B uses the purchased mailing list. A receives copies of the mailing. These records are a type of fake objects that help identify improper use of data. The distributor creates and adds fake objects to the data that he distributes to agents. We let Fi 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 many cases, the distributor may be limited in 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. The creation of fake but real-looking objects is a nontrivial problem whose thorough investigation is beyond the scope of this paper. Here, we model the creation of a fake object for agent Ui as a black box function CREATEFAKEOBJECT 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 condition. Set Ri is needed as input so that the created fake object is not only valid but also indistinguishable from other real objects. For example, the creation function of a fake payroll record that includes an employee rank and a salary attribute may take into account the distribution of employee ranks, the distribution of salaries, as well as the correlation between the two attributes. Ensuring that key statistics do not change by the introduction of fake objects is important if the agents will be using such statistics in their work. Finally, function CREATEFAKEOBJECT () has to be aware of the fake objects Fi added so far, again to ensure proper statistics. The distributor can also use function CREATEFAKEOBJECT () when it wants to send the same fake object to a set of agents. In this case, the function arguments are the union of intersection of the conditions. Implementation of CREATEFAKEOBJECT () we note that option. The function can either produce a fake object on demand every time it is called or it can return an appropriate object from a pool of objects.

C. *Data Allocation Problem*

The main focus of this paper is the data allocation problem: how can the distributor "intelligently" give data to agents in order To improve the chances of detecting a guilty agent? As illustrated in Fig. 2, there are four instances of this problem we address, depending on the type of data requests made by agents and whether "fake objects" are allowed. The two types of requests we handle were defined in Section 2: sample and explicit. Fake objects are objects generated by the distributor that are not in set T. The objects are designed to look like real objects, and are distributed to agents together with T objects, in order to increase the chances of detecting agents that leak data. We discuss fake objects in more detail in Section 6.1. As shown in Fig. 2, we represent our four problem instances with the names EF, EF, SF, and SF, where E stands for explicit requests, S for sample requests, F for the use of fake objects, and F for the case where fake objects are not allowed.



D. *Advantages Of Data Protection server*

- Secure data from unauthorized user.
- Detect guilty agent.
- Reduce amount of hackers.
- Identify guilt of agent.

III. CONCLUSION

- In this, network we are produced "data protection server" Successfully for company.
- It is working as per the user requirements.
- Using "data protection server", we can easily provide multiple level security.
- For save our confidential data from agents. So, able to provide better services in business area where needed. "Data protection server" will be used by IT professionals only.

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 Distribution 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," SignalProcessing, 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.

[11] B. Mungamuru and H. Garcia-Molina, "Privacy, Preservation and Performance: The 3 P's of Distributed Data Management," technical report, Stanford Univ., 2008.

[12] V.N. Murty, "Counting the Integer Solutions of a Linear Equation with Unit Coefficients," Math. Magazine, vol. 54, no. 2, pp. 79-81, 1981.

[13] S.U. Nabar, B. Marthi, K. Kenthapadi, N. Mishra, and R. Motwani, "Towards Robustness in Query Auditing," Proc. 32nd Int'l Conf. Very Large Data Bases (VLDB '06), VLDB Endowment, pp. 151-162, 2006.

[14] P. Papadimitriou and H. Garcia-Molina, "Data Leakage Detection,"technical report, Stanford Univ.,2008.

[15] P.M. Pardalos and S.A. Vavasis, "Quadratic Programming with One Negative Eigenvalue Is NP-Hard," J. Global Optimization, vol. 1, no. 1, pp. 15-22, 1991.

[16] J.J.K.O. Ruanaidh, W.J. Dowling, and F.M. Boland, "Watermarking Digital Images for Copyright Protection," IEE Proc. Vision, Signal and Image Processing, vol. 143, no. 4, pp. 250-256, 1996.

© 2014, IJCSMC All Rights Reserved