Detecting Data Leakage using Data Allocation Strategies With Fake objects

S.P.Subalakshmi[1], B.Geetha[2], S.P.Karthikeyan[3]
Student M.E(CSE),Mailam Engineering College,Mailam,India [1]
Assistant Professor, Mailam Engineering College,Mailam,India [2]
Student B.E(CSE),Thangavelu Engineering College,Chennai,India[3]

Abstract—With huge amount of data being exchanged these days for the purpose of business and other similar requirements, there are possibilities of data loss. A number of employees in the IT sector are also not aware and do not bound to safety compliances sometimes. Also it is seen that in organizations like a BPO huge amount of data is outsourced to third party vendors, thus data getting leaked is another possibility. The aim of this project is to protect data from getting leaked to any unauthorized person and identify the leaker. If there is a user who is getting the product will contact the distributor's website, where he needs to fill the registration form. The distributor allots the software Gmail Reader to agents with a fake object, which is an xml file; it is circulated as a zipped file. The xml file stores all the details about the user and in case if the software gets leaked to any unsanctioned user, the details will be verified before accessing the software. If the details do not match then an email is sent to the distributor, thus the access will be denied to the unauthorized person

Index Terms—Allocation strategies, data leakage, data privacy, fake records, leakage model.

I.INTRODUCTION

Latest business functions depend on email exchange for the purpose of work and business. Due to this email or data exchange the leakage has become more and easy. There is a huge destruction and wrong abuse of important data, which in turn causes damages to an organization. Whenever there is data distributor who has authentic information then the sensitive data is given to third party agents for work purpose, or it can be even outsourced. If the data is circulated to third parties and if it is available on a public domain or a private domain, then discovering the third party who has done the crime becomes an unimportant task to the distributor.

One company having partnership with another company may require allotting customers data. Also there may be another organization, which may subcontract its data to different organizations. Thus the proprietor of the data is also known as the supplier and the trusted third parties are also called as agents. Thus the aim is to identify while the data has been uncovered so as to understand the agent that has disclosed the data. There is another scenario where the actual important data cannot be bothered. Also there is a method perturbation where the data will be altered and the data will be made less sensitive, prior to providing it to the agents.

In this thesis the distributor is sending products or software's to the registered trusted third parties in a website. The trusted third parties should check the webpage and they need to fill the registration form only then they will be able to access to download the products. Also there are possibilities that the trusted source might leak the products to some other parties. Thus to avoid the data leakage from getting leaked or to have a data leakage protection model a “fake object” in the form of an xml file will be attached in the coding track. In this project the administrator is distributing the **Gmail reader software** with the xml file in the coding track. In case if any third party try to accesses the software then the details will be saved in the xml file and verify the current details with the pre-registered details and at the time of mismatching the indication will be sent to the administrator that the data has got leaked and the access will be denied.

OBJECTIVES

To identify the culprit who has leaked the data below are the objectives that are used to execute the implementation in this project.

1.The Network administrator has sensitive data where he distributes the data to some group of trusted third parties who are called as agents, where there is a possibility of some data getting leaked and being found in some other unauthorized places.

2.The administrator should identify where the data has got leaked from one or more agents. Therefore a data allocation policy is proposed to raise the chances of identifying leakages.

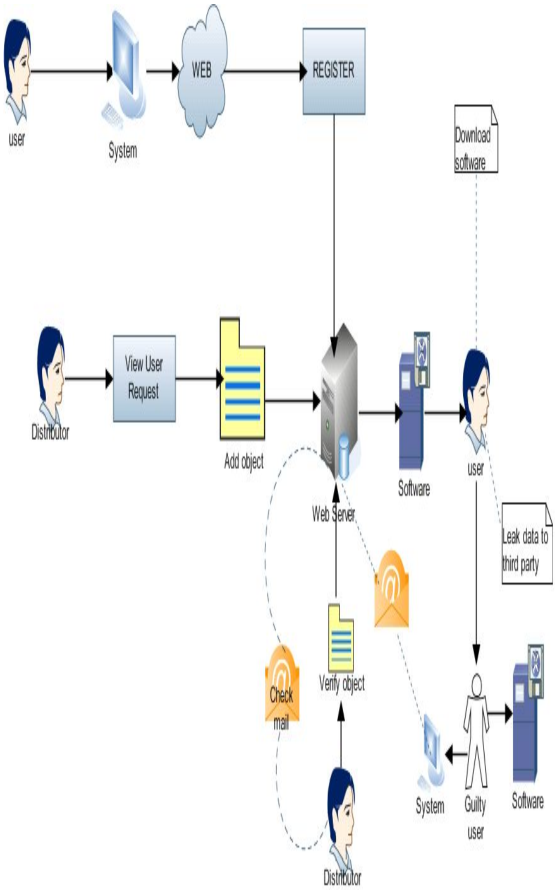
3.An object (xml file) is inserted with the data to increase the chances of tracing the data leakage and the culprit user. The XML file has a registration form where the agent has to fill all the details.

4.Our 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 a way that improves our chances of identifying a leaker with Data allocation startegies. Finally, we also consider 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.

II DESIGN AND IMPLEMENTATION

Network Administrator is also called, as the distributor is the one providing software or products to different agents or authorized users. The sanctioned users fill a registration form where they are supposed to fill in all the details about name, age organization, email ID and so on. The distributor is sending products or software’s to trusted third parties in a website. Trusted third parties should visit the webpage and they need to fill the registration form (register their name, id) and then they download the products or software’s. It might happen the trusted parties may leak the products to some other parties. To avoid the data from getting leaked or to have data leakage protection a “fake object” which is an xml file will be attached in the coding track. In this project the administrator is distributing the**Gmail reader software** with the xml file in the coding track. This xml file is used to keep saving the registered details such as name, age, email id, etc. of the trusted third parties material. In the due course the stored xml file (which has stored data about name, user ID, Email ID) will be inoculated to the transferring products. Once the downloading of the products is done, the trusted third parties try to install the software in their computer system. During this time before the installation again a registration form will appear and ask the trusted third party to fill their details. After filling the details such as (user ID, Email ID, etc) the third party user can be able to access if the provided details are exactly matched with the registered details. If the reliable third parties send the product to some other unauthorized users, during the time of installation the registration page will pop open. In that the person is required to register his details. In this period the XML file will match the registered details with the present details. If it is mismatched then the misused file that was generated is forwarded to the network administrator and the access will be denied.

Architecture Diagram



A. AGENT REGISTRATION

When an agent requires data from the company they will have to register and will attain an agent ID which is also called as the trusted agent. This method of providing details is used so as to attain access to a website. The agent registration is to attain all the details about people using the website.

The diagram below shows the agent visiting the website has to fill all the details before accessing website.

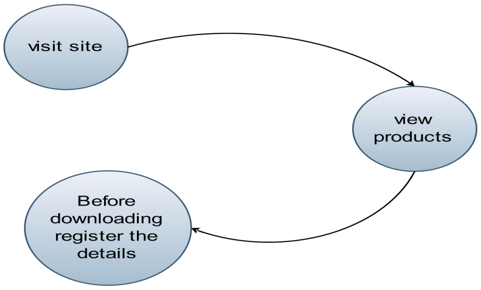


Fig: New User visiting the website

B.DATA ALLOCATION STRATEGY

Data allocation is a complex issue; as to how can the distributors cleverly provide data without the possibilities of any leakage, and increasing the possibility of identifying the culprit. With respect to the data allocation policies, this work is very similar to the watermarking which is utilized as a manner of setting ownership rights for distributed data.

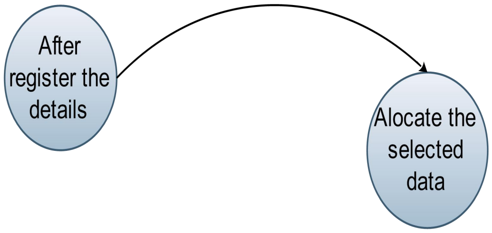


Fig: Allocating the data

Explicit Data Requests

In problems of class EF, the distributor is not allowed to add fake objects to the distributed data. So, the data allocation is fully defined by the agents' data requests. Therefore there is nothing to optimize.

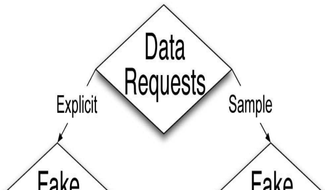


Fig: Leakage Instances

The distributor cannot remove or alter the R1 or R2 data to decrease the overlap R1 ∩ R2. However, say that the distributor can create one fake object (B =1) and both agents can receive one fake object (b1 = b2= 1). In this case, the distributor can add one fake object to either R1 or R2 to increase the corresponding denominator of the summation term. Assume that the distributor creates a fake object f and he gives it to agent R1. Agent U1 has now R1 = {t1,t2,f} and F1 ={f} and the value of the sum-objective decreases to 1:33 < 1:5.

If the distributor is able to create more fake objects, he could further improve the objective. We present in Algorithms 1 and 2 a strategy for randomly allocating fake objects. Algorithm 1 is a general "driver" that will be used by other strategies, while Algorithm 2 actually performs the random selection. We denote the combination of Algorithm 1 with 2 as e-random. We use e-random as our baseline in our comparisons with other algorithms for explicit data requests.

Algorithm 1. Allocation for Explicit Data Requests (EF)

```
Input: R1, . . .,Rn, cond1, . . . , condn, b1, . . . , bn, B
Output: R1, . . .,Rn, F1, . . . , Fn
1: R ← ∅  Agents that can receive fake objects
2: for  i =1, . . . , n do
3: if bi > 0 then
4: R ← R  ∪ {i}
5: Fi ← ∅
6: while B > 0  do
7: i ← SELECTAGENT (R,R1, . . .,Rn)
8: f ← CREATEFAKEOBJECT(Ri, Fi, condi)
9: Ri ← Ri ∪ {f}
10: Fi  ← Fi ∪ {f}
11: bi  ← bi - 1
12: if bi = 0 then
13: R ← R\{Ri}
14: B ← B _ 1
```

Algorithm 2. Agent Selection for e-random

```
1: function SELECTAGENT (R,R1, . . .,Rn)
2: i ← select at random an agent from R
3: return I
```

In lines 1-5, Algorithm 1 finds agents that are eligible to receiving fake objects in O(n) time. Then, in the main loop in lines 6-14, the algorithm creates one fake object in every iteration and allocates it to random agent. The main loop takes O(B)time. Hence, the running time of the algorithm is O(n + B).

Theorem: Algorithm e-optimal yields an object allocation that minimizes both sum- and max-objective in problem instances of class EF.

Sample Data Requests

With sample data requests, each agent Ui may receive any T subset out of different ones. Hence, there are different object allocations. In every allocation, the distributor can permute T objects and keep the same chances of guilty agent detection. The reason is that the guilt probability depends only on which agents have received the leaked objects and not on the identity of the leaked objects.

Note that the distributor can increase the number of possible allocations by adding fake objects (and increasing

|T|) but the problem is essentially the same. So, in the rest of this section, we will only deal with problems of class SF, but our algorithms are applicable to SF problems as well.

Algorithm 3. Allocation for Sample Data Requests (SF)

```
Input: m1, . . .,mn, |T|.   Assuming mi _ |T|
Output: R1, . . .,Rn
a ← 0|T| . a[k]:number of agents who have
     received object tk
R1 ← ∅ . . .,Rn← ∅ ;
remaining ← mi
while remaining > 0 do
for all i =1,. . . ,n : |Ri| < mi do
k ← SELECTOBJECT(I,Ri) . May also use
   additional parameters
Ri ← Ri ∪ {tk}
a[k]← a[k]+ 1
remaining ← remaining – 1
```

Algorithm 4. Object Selection for s-random

```
function SELECTOBJECT(i,Ri)
k select at random an element
return k
```

C.OBJECT MODULE

Fake object (xml file) developed by the distributor are attached to the software so as to raise the possibilities of identifying culprits who leaks the data. Therefore the distributor is enabled to include fake objects to the circulated data in order to increase the possibility of tracing the culprit agents. The idea of utilizing the fake object has been stimulated by using the track reports in the mailing lists. Once the data is allocated, an XML file is added to the product or the software.

The block diagram below shows that after allocating the selected data, the xml file is added to the product/software/data.

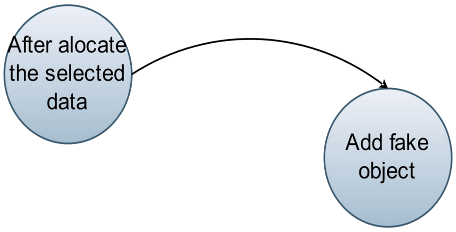


Fig: Insertion of fake object

D.OPTIMIZATION

This Module will be the administrator's data distribution to the agents where there can be some restraint or an objective. The restraint is that: The distributor's restraint is to persuade the requests of the agents, by specifying them by way of the amount of objects they ask otherwise by way of all accessible entities that please the scenario. The goal is to identify the trusted third parties who are called as agents who leak the section of sensitive data. A fake object is considered while data distribution, which is the only moderation to the constraint.

With the addition of the xml file to the product/software/data and integrating it together to form a single zipped file.

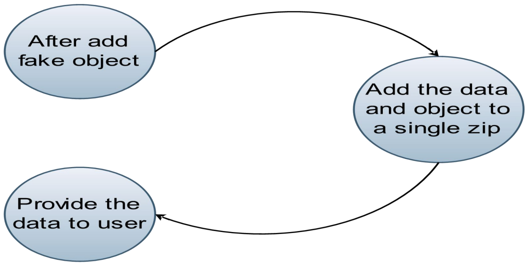


Fig: Providing zipped file to the agents

E. DATA DISTRIBUTOR:

A data distributor circulates a lot of sensitive data to a group of authorized users. If the data has got leaked and is present on the website or at any unsanctioned place like a pen drive, personal computer, then the user's data will be at risk. The distributor should calculate the possibility where the leaked data has come from one of the agents, as opposite to partaking individually by some extra source. The distributor develops and inoculates the fake element, which is an xml file to the data that he circulates to agents. The fake objects should be developed carefully so the agents will not be able to differentiate the same from the actual object, in this case the Gmail reader software.

The block figure below shows dispensing the data and with the support of the xml file one can identify the malevolent agent.

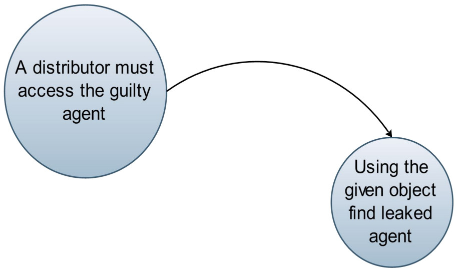

A distributor must access the guilty agent
Using the given object find leaked agent

Fig: Tracing the guilty agent

DATA FLOW

The flow chart below explains the complete plan of the project. Where a user initially contacts the supplier for any service or a product, he needs to fill the registration form. Once the user gets registered the distributor can allocate data to him. The user will download the software to check it where he gets permission to use the software, now if a user forwards this software to unknown person, the new user will try to download the software and will have to fill the form, where these details are saved in as an xml file, this gets forwarded to the distributor, where the distributor come to know about the person who has leaked the data and the culprit can be tracked.

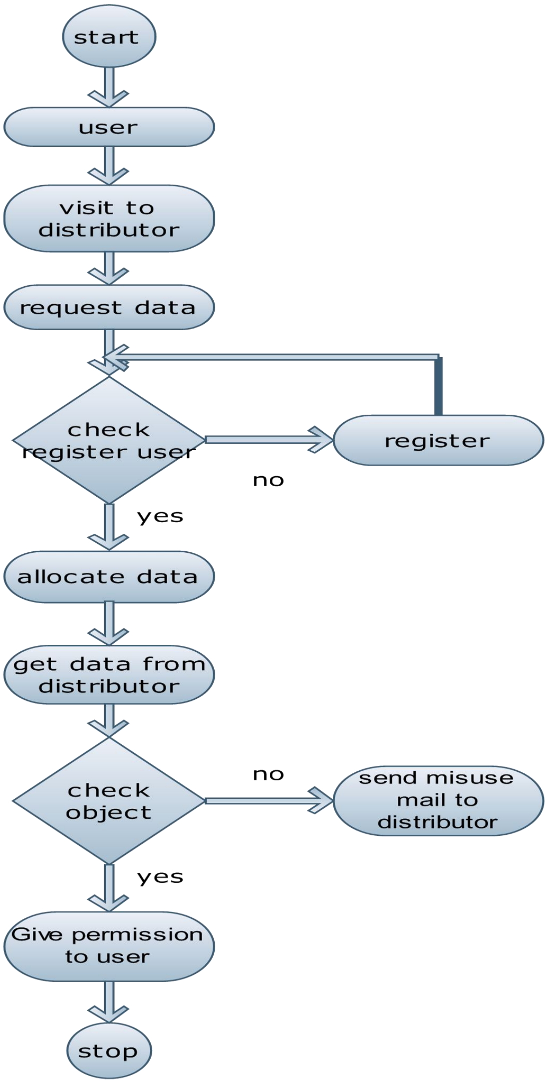


Fig: Work Flow

III EXPERIMENTAL RESULTS

A.CRITICAL ANALYSIS

Addition of the XML file, which is a fake object to the distributed data, is done. Sometimes these elements do not resemble to the actual entities but they seem to be genuine to the agents.

The XML file functions as watermark for the complete set, without any modification to the single members. The data allotment methods for different agents by the distributor enhance the possibility of detecting the leakage.

The use of the fake XML element resemble to the actual entities, seem to be real to the agents. The fake XML element seems to function as a watermark for the complete set with no altering several single members. In such a case if the agent was provided with single or additional fake data which are leaked, then the distributor can be very sure that the agent was guilty.

B.FEASIBILITY

This project is executed to verify the lucrative influence data leakage protection will have on the organization. The amount of funds that the companies have for research and development are a little less. Any system formed should not have huge demand on the present technical sources. This will result to high demand on the present technical resources. This will result in huge demand for the client.

C.NORMALIZATION

Having the best possible table formats, extracting data entries, accompanied by careful investigation of different kinds of information with each other is done. In the area of the relational database design, normalization is a methodical manner of making sure that the database is appropriate for a given reason with query and free of a given unwanted features- like the addition, renovate and obliteration irregularities that results in loss of the honesty of the data.

CONCLUSIONS

In today's world, there will be no requirement to give important data to the agents that might not knowingly leak data to other agents. Also if one has to provide sensitive data to any agent, a watermark has to be inserted with every element, so as to track the roots with the complete data. In a number of scenarios there are various works that cannot be reliable completely and one cannot be sure if the leaked data has arrived from trusted third parties who are called as agents or from some different resource, because data cannot disclose watermarks. Therefore also the use of xml file, which stores data of different agents, helps in tracing the agent that has leaked the data. Access is not granted to unauthentic user, a mail is generated and sent to the distributor if the software Gmail Reader reaches a non-sanctioned user. The registration form filled helps in verifying if the user is authentic or not.

The methodology used in the project is executed in a manner that a wide data distribution policy can enhance the distributor's possibilities of detecting the guilty. The circulating elements can in a manner astutely be made suggestive in detecting the culprit or the leaker , where there is a huge overlay in data circulated that the agents achieve.

REFERENCES

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

- Alzheimer Europe (2009). "The four main Approaches-Types of Research." Available at http://www.alzheimer-europe.org/Research/Understanding-dementia-research/Types-of-research/The-four-main-approaches

- Apache XML Project. [Available Online at: http://xml.apache.org/]

- Bonatti, S.D.C. di Vimercati, and Samarati (2002), "An Algebra for Composing Access Control Policies," *ACM Trans. Information and System Security, vol. 5, no. 1, pp. 1-35, 2002.*

- Elovici, Shabotai, Rokach (2013) "A Survey of Data Leakage Detection and Prevention Solutions,"*5 Springer Briefs in Computer Science, DOI 10.1007/978-1-4614-2053-8_2; 2013 VIII, 92 p.9.*

- JAGTAP, PATIL. AND ADHIYA K.P. (2012) "Implementation of Guilt Model with Data Watcher for Data Leakage Detection System" *Advances in Computational Research ISSN: 0975-3273 & E-ISSN: 0975-9085, Volume 4, Issue 1, 2012.*

- Ruanaidh, Dowling, and Boland (2006), "Watermarking Digital Images for Copyright Protection," *IEEE Proc. Vision, Signal and Image Processing, vol. 143, no. 4, pp. 250-256, 2006.*

- XML Encryption Syntax and Processing, W3C Recommendation (2002). [Available Online at: http://www.w3.org/TR/xmlenc-core/]