A Novel Model for Data Leakage Detection and Prevention in Distributed Environment

S.Praveen Kumar, Dr.Y.Srinivas, Dr.D.Suba Rao, Ashish Kumar

***Abstract*— Data Leakage can be referred as the unauthorized transfer of classified information from a computer or data centre to the outside world. Big data is an evolving term that describes any voluminous amount of structured, semi-structured and unstructured data that has the potential to be mined for information. The data loss in such huge massive data sets may cause critical vulnerabilities. In this paper we proposed a novel model in Hadoop master/slave architecture for data leakage prevention with LRA algorithm. The LRA algorithm evaluates the Least Reliable Agent and sends it to the DLA (Data Leakage Avoider). DLA prevents the NameNode from allocating data to these agents.**

***Index Terms*— Data Leakage, Big Data, Hadoop Architecture, LRA, DLA.**

I. INTRODUCTION

In modern business process sometime sensitive data is distributed to set of supposedly trusted agents (third parties) by the owner of data. If in case distributed data where found at unauthorized place (e.g. on the website or somebody's laptop) then the owner must be able to assess the likelihood that the leaked data come from one or more agents or not. Organizations mainly apply data or information security only in terms of network protection from intruders and hackers, but due to globalization and digitization there is rapid growth in the amount of sensitive data processing applications. In the organizations these data can be accessed from different medium which increases the chances of data leakage and trouble in guilt assessment. Data allocation strategy includes fake records in the distribution set along with sensitive data. It acts as watermark to identify the corresponding owner. If such sensitive information is leaked then with the help of fake records and distribution logic involvement of agent in the leakage can be traced. Suitable data allocation strategy improves the probability of guilt detection and this reduces the catastrophic effect of data leakage [1].

Data mining is the extraction of hidden, predictive information patterns from large data bases [11]. It is used to find the relevant and useful information from data bases. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Sometimes sensitive data is leaked and found in unauthorized places. For example, 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. The owner of the data is called the distributor and the supposedly trusted third parties are called the agents. 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.

The term and use of big data is nothing new. In fact, more and more companies, both large and small, are beginning to utilize big data and associated analysis approaches as a way to gain information to better support their company and serve their customers. Big Data can reduce the processing time of large volumes of data in the distributed computing environment using Hadoop. It also can predict potential cyber security breaches, help stop cyber attacks, and facilitate post-breach digital forensic analysis.

Security and privacy concerns are growing as big data [10] becomes more and more accessible. The collection and aggregation of massive quantities of heterogeneous data are now possible. Large-scale data sharing is becoming routine among scientists, clinicians, businesses, governmental agencies, and citizens. However, the tools and technologies that are being developed to manage these massive data sets are often not designed to incorporate adequate security or privacy measures, in part because we lack sufficient training and a fundamental understanding of how to provide large-scale data security and privacy. We also lack adequate policies to ensure compliance with current approaches to security and privacy. Furthermore, existing technological approaches to security and privacy are increasingly being breached, whether accidentally or intentionally, thus necessitating the continual reassessment and updating of current approaches to prevent data leakage. In general, big data is stored in a distributed environment; therefore, there are security threats from networks. Data security issues in cloud storage include personal privacy protection, data security protection, intellectual property protection, and commercial secrets and financial information protection, etc. In a Data-as-a-service (DaaS) environment, data unavailability or data loss is a high risk thing. New methods and models should be created to handle the security risk of data that resides in cloud. New encryption approaches should be developed or new processes should be defined to separate core vs. noncore data to reduce the security risk of hosting data in cloud.

Big Data analytics extracts and correlates data, which makes privacy violation easier. Methods need to be developed to minimize privacy invasions during Big Data analytics. Abuse of big data stores should be prevented. It is necessary to secure big data stores and produce documents on security in cloud computing to secure big data. Big data provenance is another challenge. Because Big Data allows for expanding data sources, data can be from different sources. The integrity, authenticity or trustworthiness of each data source should be verified [12].

S.Praveen Kumar, Assistant Professor, Department of IT, GIT, Gitam University
Dr.Y.Srinivas Ph.D, Professor & HOD, Department Of IT, GIT, Gitam University
Dr.D.Suba Rao, Phd Professor & Head School Of Engineering & Technology, Centurion University
Ashish Kumar, B.Tech [IT], Department of IT, GIT, Gitam University

II. Data Allocation Problem:

2.1. Fake Objects:

The distributor may be able to add fake objects to the distributed data [2]. In order to improve the effectiveness in detecting guilty agents [3]. However, fake objects may impact the correctness of what agents do, so they may not always be allowable. 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. Depending upon the addition of fake tuples into the agent's request, data allocation problem is divided into four cases as [5]:

i. Explicit request with fake tuples (EF)
ii. Explicit request without fake tuples (E~F)
iii. Implicit request with fake tuples (IF)
iv. Implicit request without fake tuples (I~F)

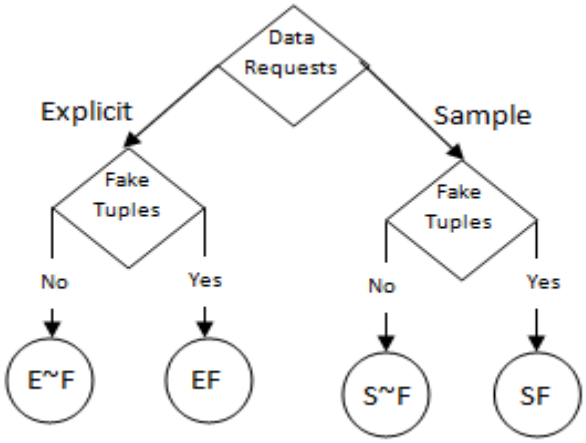

Fig.2.1. Leakage Problem Instance

2.2. Data Allocation Strategies:

In this section we describe allocation strategies that solve exactly or approximately the scalar versions of approximation equation. We resort to approximate solutions in cases where it is inefficient to solve accurately the optimization problem.

2.2.1. Explicit Data Requests:

In case of explicit data request with fake not allowed, the distributor is not allowed to add fake objects to the distributed data. So Data allocation is fully defined by the agent's data request. In case of explicit data request with fake allowed, the distributor cannot remove or alter the requests R from the agent. However distributor can add the fake object. In algorithm for data allocation for explicit request, the input to this is a set of request from n agents and different conditions for requests. The e-optimal algorithm finds the agents that are eligible to receiving fake objects. Then create one fake object in iteration and allocate it to the agent selected. The e-optimal algorithm minimizes [5] every term [6] of the objective summation by adding maximum number of fake objects to every set yielding optimal solution.

2.2.2. Sample Data Requests:

With sample data requests, each agent Ui may receive any T subset out of 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. The distributor gives the data to agents such that he can easily detect the guilty agent in case of leakage of data. To improve the chances of detecting guilty agent, he injects fake objects into the distributed dataset [5]. These fake objects are created in such a manner that, agent cannot distinguish it from original objects. One can maintain the separate dataset of fake objects or can create it on demand. In this paper we have used the dataset of fake tuples [6].

III. Fake Object Record Table:

Fake object record table is used to store information regarding the agents and the corresponding number of fake objects allocated to it. Using this record we can detect whether there is a leakage or not. We compare the current number of fake objects an agent has along with the total number to fake objects initially allocated to it. The following table gives the example of the Fake object record table giving some random values.

S. No.	Agent Number	Number of Fake objects allocated
1.	1001	10
2.	1002	9
3.	1003	12

Fig. 3.1. Example for Fake object record table

Every agent has an agent number and all the fake objects given to a particular agent are allocated with a unique id. Therefore the object record table will be verified and the frequent data leak of the object and their corresponding agents who leaked it can be assessed.

IV. Proposed Model for Data Leakage Prevention In a Distributed Environment (Big Data):

4.1. Least Reliable Agent [LRA] Algorithm:

We consider a time counter variable 'T' and initialize it to zero. The variable 'SUB' stores the difference between the numbers of fake objects allocated to an agent and the number of fake objects it actually has at that instance of time. 'RATE' is a variable which is used to store the rate at which the data is leaking. DATA is a structure node which represents the Fake object record table and stores the agent number and the number of fake object allocated to it. ARR is an array which stores the fake object for the agent. LEAST_REL is the agent which is evaluated to be the least reliable one. Using this algorithm we can detect the future reliability of an agent and take necessary measures.

The algorithm is as follows:
T=0; //Time Counter is set to 0

```
For all i = 1 . . n do { /* n refers to the total number of agents present */
for all T=1..m do { /* m refers to the total amount of time you want to perform this check*/
if(( DATA[i]→AGENT→FAKE) = ARR.length )
/*Currently how many obj
                are present with the agent*/
 return;
else {
SUB = ((DATA[i]→AGENT→FAKE) – ARR.length);
/*SUB is how many no. of lost objects for the corresponding agent.*/
RATE[i]= SUB / T;   /* rate at which the
                       data is leaking*/
}
T += 2;
}
if ( RATE[i] > Rate [i-1])
LEAST_REL = i;
}
```

4.2. Agent reliability check in Hadoop framework (Data Leakage Prevention):

Hadoop architecture [10] is used for unambiguous storage and retrieval of huge amount of structured data, unstructured data or semi structured data. This is done using HDFS [7] (Hadoop Distributed File System) and Hadoop MapReduce [8]. While handling this huge amount of data it is required to keep a check on how and where the data is leaking and take steps to stop that leakage. In this section we provide a model to prevent data leakage. Figure 4.1.gives the diagrammatical representation of the model. We introduce a Data Leakage Avoider (DLA) at the master node of the Hadoop architecture along with NameNode and JobTracker and a Reliability checker (RC) at each slave node along with DataNode and TaskTracker.

4.2.1. Reliability checker (RC):

Reliability checker evaluates the least reliable agent and sends the information to Data Leakage Avoider which prevents the NameNode from sending data to those agents. The reliability checker consists of the LRA algorithm mentioned above 4.1 for evaluating the least reliable agent. This algorithm helps in finding the rate at which the data is leaking and thus finds the agent which is responsible for the maximum data leakage.

4.2.2. Data Leakage Avoider (DLA):

This is used to store the information related to each slave node regarding the least reliable agents present in the DataNodes. The Reliability Checker sends status of the least reliable agent from that particular slave node to the DLA. Based on this information about the least reliable agent the NameNode prevents allocating data to that agent. This process is frequently done after some time interval. This helps in reducing the chances of data leakage to huge extends.

4.2.3. Working:

Reliability Checker and Data Leakage Avoider are introduced at each slave nodes and master node respectively. The Reliability Checker evaluates the least reliable agent from the DataNode based on the rate at which data is being leaked. This information is send to the Data Leakage Avoider for every fixed amount of time (e.g. 5 minutes). This helps the DLA to store the information regarding the least reliable agents in which data must not be allotted. The DLA prevents the NameNode from sending data to those agents. If the maximum number of agents present in the DataNode is labelled as least reliable, then no data must be sent to that DataNode.

4.2.4. Model Implementation:

The following diagram shows the implementation of the above concept:

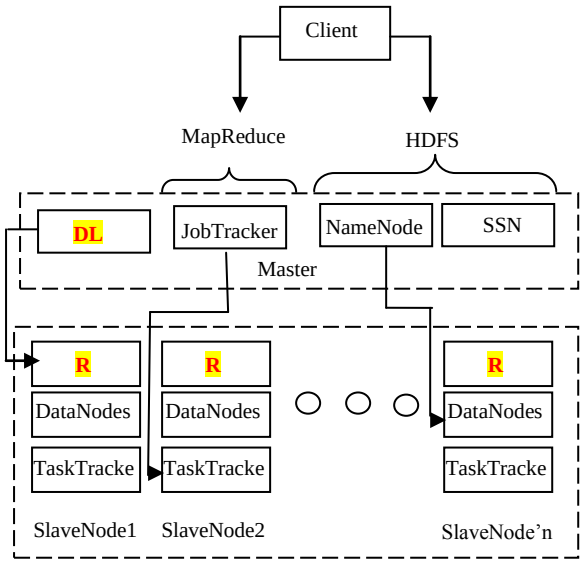

Fig. 4.1. Model Implementation using DLA and RC in Hadoop Architecture.

V. Conclusion:

Data leakage has been a major threat to data security and integrity. To overcome this problem various data allocation strategies have been proposed and implemented. In this paper we have focused on the data leakage problem in a distributed environment (big data). We proposed concept for data leakage prevention using a Least Reliable Agent algorithm and a model implementation using Data Leakage Avoider and Reliability Checker. This model helps to decrease data leakage to high extents by avoiding the NameNode to allocate data to the agents which are labelled as the least reliable. Our model is a simple prototype designed in order to reduce data leakage. The LRA algorithm evaluates the least reliable agent from a set of agents by calculating the rate at which the data is leaking. We provided a novel model in Hadoop architecture. In future there is a scope to work on better data leakage detection algorithms which would be relevant to our model and also provide efficient detection.

References:

[1] Data Leakage Detection by using Fake Objects By Rama Rajeswari Mulukutla& P. Poturaju https://globaljournals.org/GJCST_Volume13/5-Data-Leakage-Detection.pdf

[2] Data Leakage Identification and Blocking Fake Agents Using Pattern Discovery Algorithm http://www.rroij.com/open-access/data-leakage-identification-and-blockingfake-agents-using-pattern-discoveryalgorithm.pdf

[3] Data Leakage Detection http://www.ijarcce.com/upload/november/14-Data%20Leakage%20Detection.pdf

[4] http://www.slideshare.net/OcularSystems/data-leakage-detection-13979738
[5] Data Leakage Detection http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.665.9894&rep=rep1&type pdf
[6] Data Leakage Detection http://ijcsmc.com/docs/papers/May2013/V2I52013108.pdf
[7] HDFS https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
[8] MapReduce https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
[9] Big Data http://www.sas.com/en_us/insights/big-data/what-is-big-data.html
[10] A Complete Introspection on Big Data and Apache Spark –S.Praveen Kumar,Dr. Y.Srinivas, Dr. D.Subba rao ,Ashish Kumar—http://www.ijsdr.org/papers/IJSDR1604006.pdf
[11] An Introduction to Data Mininghttp://www.thearling.com/text/dmwhite/dmwhite.htm
[12] What is Big Data Analytics? http:/www/01.ibm.com/software/data/infosphere/hadoop/what-is-big-data-analytics.html
[13] Big Data (introduction) http://bigdata.ieee.org/
[14] A Model for Data Leakage Detection Panagiotis Papadimitriou 1 , Hector Garcia-Molina 2 Stanford University 353 Serra Street, Stanford, CA 94305, USA
Development of Data leakage Detection Using Data Allocation Strategies
http://cse.final-year-projects.in/a/275-development-of-data-leakage-detection-using-data-allocation-strategies.html
[15] Observing and Preventing Leakage in MapReduce http://research.microsoft.com/pubs/255857/MSR-TR-2015-70.pdf

S.Praveen Kumar, Assistant Professor, Department of IT, GIT, Gitam University

Dr.Y.Srinivas Ph.D, Professor & HOD, Department Of IT, GIT, Gitam University

Dr.D.Suba Rao, Phd Professor & Head School Of Engineering & Technology, Centurion University

Ashish Kumar, B.Tech [IT], Department of IT, GIT, Gitam University