Privacy Breach Detection and Hindrance of data in Cloud Computing

Shivakumara T[1], Dr. Rajshekhar M Patil[2]

[1]Assistant Professor, Department of MCA, BMS Institute of Technology and Management, Bangalore, Karnataka, India, shivakumarat@bmsit.in

[2]Professor, Department of CSE, Amruta Institute of Engineering and Management Sciences, Bangalore, Karnataka, India, pvsmr1@gmail.com

Abstract:

In the recent years Data Leakage Detection as a main problem in Cloud Computing, which has heterogeneous users. The increasing ability to trail and collect large amounts of data with the use of technology has lead to an interest in the development of data protection algorithms, which helps to preserve user sensitive data's in the distributed environment. There are several Data Leakage Detection and Data Leakage Prevention techniques have initiated for securing and protecting data's. A recently proposed technique addresses the issue of data leakage in transferred data's by sampling and matching techniques. However, the method can only perform detection part within the application level. There is several other ways to leak the transferred data within the host rather than the application. The proposed method designed and developed a framework for distribution data security, which have developed in an email server for reconstruction secure email transfers. This includes different types of algorithms to detect and prevent data leaks either partially or fully. This is more effective than the currently available method in terms of the level of information loss. The proposed approach performs prediction and probability finding in order to identify and protect data from leakers. The experimental result shows the effective results on the distributed data in email security.

IndexTerms **- Information Security, Data Leakage Detection, Data Leakage Prevention.**

I. INTRODUCTION

Organizations and people are increasingly concerned about confidential data [1] being leaked to the public. Managing Security and reduce the risk is very vital in todays fast running world [10]. Traditionally, security processes such as information security rules, as well as traditional security methods like as firewalls, virtual private networks, and intrusion detection systems, have been used to ensure confidentiality. [2][3]. Moreover, these methods lacked proactivity and attention to preserving personal data, and therefore necessitated the use of established rules. As confidential data might exist in many formats in various leak routes, this can have catastrophic consequences. [4]. As a result, there has been a push to alleviate these disadvantages through more efficient procedures. Data Leakage Prevention Systems (DLPSs) are specific systems for detecting and preventing sensitive data leakage in use, transit, and at repose. [5]. To detect and prevent data leakage, DLPSs employ a variety of approaches to assess the content and context of confidential data. Although IT security vendors and researchers are increasingly designing and developing DLPSs as separate products, the term remains vague. A data leakage detection and prevention technique for email security is proposed in this paper. This paper Straight forwardly defines proposed DLP in the distributed heterogeneous data sharing networks. Nowadays data publishing is a common process. However, the issue of data sharing is difficult when the data is more sensitive and secure. When data is spread across multiple entities, there is a greater risk of sensitive material being leaked and accessed in unauthorized places. The data should be safe and secure from unauthorized users. Data distributor affected a lot because of the data leakage problem. Some simple and recent research and news show the authorized users have shared the secret data with some other unauthorized users. Therefore, improving the privacy and identifying the data leaker with proof is the main aim of the proposed system. For example, a corporation may have joint ventures with other businesses that necessitate the sharing of client information Furthermore, a technique for allocating items to agents in such a way that enhances the probability of discovering a leaker has been given. The Objective of the current proposal is to achieve a complete data security and applying for rule-based data protection from data leakage in the distributed environment. The proposed system aims at protecting the data against an adversary who has the knowledge and partial rights of at most m items in a specific transaction and also provide a cost-effective non-cryptographic data protection against private data-sharing and rule-based data leakage protection and leaker detection.

Protecting owner information is also an important goal, which performed by designing an effective algorithm to ensure better protection against data disclosure. The proposed system also aims at developing a new method, which has the ability to detect and prevent partial or full data leakage in the email server. Another purpose is to detect when a distributor communicates sensitive data that could be leaked by a user through other programmers. In this case, the detection of partial or full leaked data will be more helpful. The proposed system aims to detect the leaker when they try to leak the whole or a part of file of the owners. By using prediction algorithms, you can increase your odds of finding a fraudulent agent that leaks all of his data items.

II. PROBLEM DEFINITION

In the literature, a set of DLD and DLP techniques were proposed [6]. A Content Inspection Technique was recently presented to detect sensitive data leaks in the content of files or network traffic. The Data Leak Detection method is analyzed by comparing the similarity of two sampled sequences. The Sequence Alignment Technique [7] is used for Data Leakage Detection and Complex Data-leak Patterns. The technique is intended to discover sensitive data patterns that are lengthy and inexact. This is done using a comparable sampling algorithm, which compares the similarity of two sequences that have been sampled separately. This technology has a high detection accuracy when it comes to detecting transformed leaks. The Sequence Alignment algorithm can only track the application level data leakage rather than another type of Data-Movement based tracking. The existing system failed to perform both detection and prevention of data leaks. In the existing sequence alignment and sampling techniques were used to only detect the transformed data leaks. However, the application was not performed a complete data leak detection and protection. The data leakage within the application can be easily detected, where data movement based detection has not yet performed well. The existing DLD and DLP need more computations to detect and prevent data leaks.

There are several challenges associated with Data Leakage Prevention (DLP), data authorization with Partial and Full Data Leakage Detection (DLD) [8]. When data contains a leak, the Data Leakage Detection provider first learns about the vulnerable data.

The problem is determining how to authorize the data distributors and limit the amount of information that the owner can learn in the event of a data breach. The plaintext packet payload is accessible to the provider. The second problem is determining how to improve detection accuracy. It is clear from the literature that data leakage detection and prevention strategies are understudied. The majority of current solutions have significant flaws, especially when dealing with personal data that is constantly changing. This is due to the fact that they rely on rigid approaches. Though with certain robust fuzzy fingerprinting [9] and statistics, the sensitive data interpretation can be disclosed using many misrepresentations. As a result, "how to detect semantically the content of sensitive data in order to prevent data leaking" is a possible research subject in this domain. Even if it is changing, a good future DLPS should be able to semantically classify private data. Despite the fact that some academics insist on depending on semantics, Without knowing the content, it is difficult to retain semantics. DLPSs currently keep copies or references to confidential data. This makes it possible to detect leaks as soon as it occurs. Unfortunately, this is insufficient since sensitive information can be developed without adhering to categorization processes. Without the requirement to manage exact copies of existing and new data, DLPSs should be able to heuristically detect such data. Furthermore, detection approaches necessitate a thorough examination that involves content inspection and indexing.

III. PROPOSED MODEL:

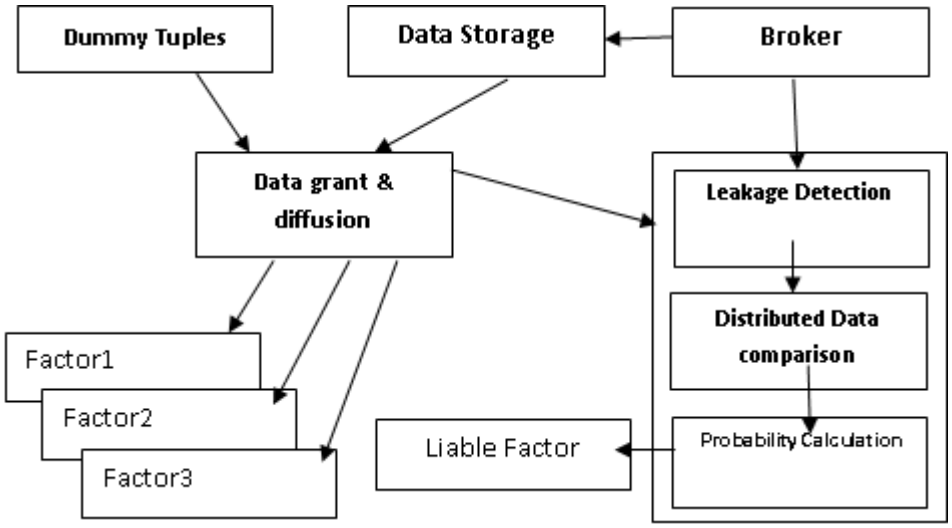

Figure 1 Proposed Architecture for Data Leak Detection

Figure 1 depicts the proposed system's overall architecture, which includes the user's initial rule specification as well as the data transmission details for Data Leak Detection and Prevention. The proposed system collects the user's activity log in online and offline and gets the rules for data protection from the users. That is described on the first phase. These details are passed to the process, which performs the detection of hidden patterns and calculates the probability of data being leaked by a user by using their log. Using the hidden factors and data movement tracking information's, the system find the data leak either partially or fully. It returns the percentage for the leaked data.

The sensitive data set K=k1, k2,... kn is owned by the distributor. The data objects are requested by the agent Ai from the distributor. The objects in K could be of any type or size, for example, tuples in a relation or database relations. Each agent receives a subset of data from the distributor, The distributor detects that a set L of T has leaked after delivering objects to agents. This indicates that L has been discovered in the possession of a third party. The agent Ai is given a subset Ri of objects T, which is determined by an Inferred or Straight forward request. Inferred Request Ri = Inferred (T, mi): Agent Ai can be provided any subset of mi entries from T. Straight forward Agent Ai receives all T objects that satisfy Condition when Request Ri = Straight forward (T, Condi).

3. 1 Distribution Module:

The distributor may be able to add false objects to the dispersed data to increase his efficacy in detecting fake corrupt agents. Fake objects, on the other hand, may have an influence on the accuracy of what agents do, therefore they may not always be allowed. Our use of fake objects was informed by the use of trace records in mailing lists. In this scenario, company A sells a mailing list to company B that will only be utilised once (e.g., to send advertisements). Company A adds trace data containing company A's addresses. As a result, When company B uses the acquired mailing list, A receives copies of the mailing. These records are a form of fictitious object that aids in the detection of data misuse. The data that

the distributor transmits to agents contains fake objects that the distributor produces and adds to it. The data allocation problem is separated into four instances based on the inclusion of fake tuples in the agent's request:

i. Straight forward request using Dummy tuples (SP)
ii. Straight forward request with no Dummy tuples (S~P)
iii. With Dummy tuples, an indirect is made (PI)
iv. Collateral request (CR) with no Dummy tuples(C~P).

Inferred Request Ri = Absolute (P, k_i): Agent A_i can be provided any subset of k_i records from P.

3. 2 Boosting Module:

The data distribution to agents by the distributor has one limitation and one goal. The distributor's constraint is to fulfill agents' requests by giving them the quantity of objects they want or all accessible objects that meet their criteria. His goal is to be able to detect any agent who releases any of his information. The purpose is to grow the likelihood of catching a guilty agent who leaks all of his data objects. The chance that an agent is corrupt is Pr {Gj|S =Ri}, or simply Pr {Gj |Ri}, if the distributor finds a leaked table S that includes all objects.

Allow n agents to submit data requests to the distributor. The distributor wants to offer agents A1,A2.....An tables R1,R2.....Rn. In order for Distribution to satisfy the agent's request, it must maximize the fault probability disparities.

Δ (i, j) for all i, j= 1, 2, ……n and i≠j.
maximize(overR1….,Rn) (…,.Δ(i,j),…) i≠j……..(A)
minimize(over R1,….,Rn) (.., $\left|Ri \cap Rj\right| \div \left|Ri\right|$,…) i≠j

3.3 Algorithm for Data Leak Detection :

Allocation of Data Straight forwardly:
Input: -

i. Data set for Distribution K= {k1,k2,k3, …. kn}
ii. R- The agent's request
iii. Cond- The agent's condition.
iv. z= the number of tuples assigned to an agent z<n, which are chosen at random.

Output: - O – Sent Data
Method :

1. O=0, k^1=0
2. For i equal to -1 less than n do
   If(k.corral equal context) then
3. K^1=K^1 Union {ki}
4. For i=0 to i<z do
5. O=O Union {ki}
6. K^1=K^1-{ki}

7. If K^1equals to 0 then
8. Go in to the 2 step.
9.Assign dataset K to a certain agent.
10. Process is repeated for each agent.

IV. RESULTS AND DISCUSSION

We used a bunch of 500 objects in our scenarios, and every agent's requests were accepted. Since we are analyzing their trust values, there is no limit to the number of agents. The following is the workflow of our system:

1. Straight forward or Inferred request from the agent.
2. The system was provided a leaked dataset as an input..
3. The list of all agents with common tuples with leaked tuples is collected, and the appropriate guilt probabilities are determined.
4. It demonstrates when the overlaps with the disclosed dataset shrinks, the probability of locating the guilty agent increase. The fundamental approaches for leak detection systems in a variety of fields, as well as offering a multi-angle strategy to dealing with situational concerns, were all properly studied.

When sensitive data is transferred, each object should be watermarked so that its origins can be traced with complete accuracy. If specific data cannot accept watermarks, the likelihood that an agent is accountable for just leak can be determined based on the intersection of the data with the leaked data, as well as the probability that objects can be inferred by any other methodology.

Case 1 : In this case, the number of tuples assigned to an agent assigned for z which is greater than k where $z>[k], z=\Sigma i=1\ldots n$

Factors	Folder Seek	Folders set
Framework1	8	8
Framework2	7	0
Framework3	8	5
Framework4	6	0

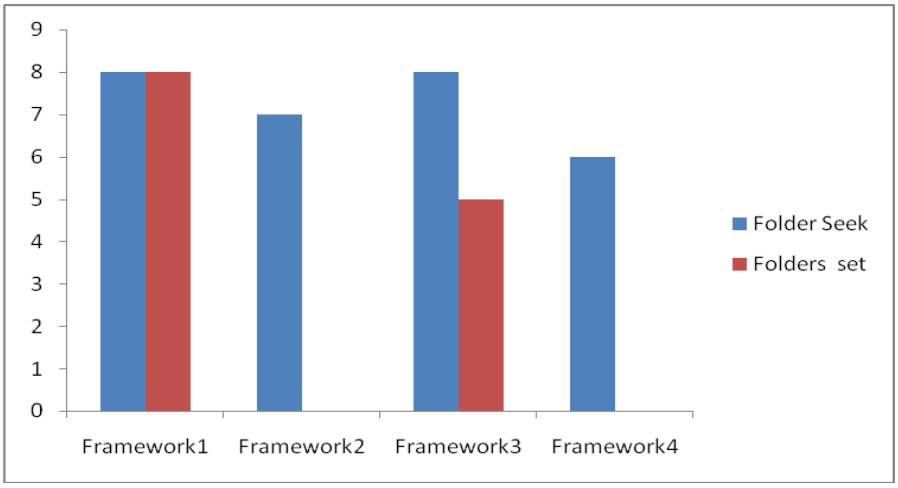

Figure 2 Graph shows number of requested and Allocated files

Case 2 : In this case, the number of tuples assigned to an agent assigned for z which is lesser than k where z<[k],z=Σi=1…n

Factors	Folder Seek	Folders set
Framework1	8	8
Framework2	7	-
Framework3	8	5
Framework4	6	-

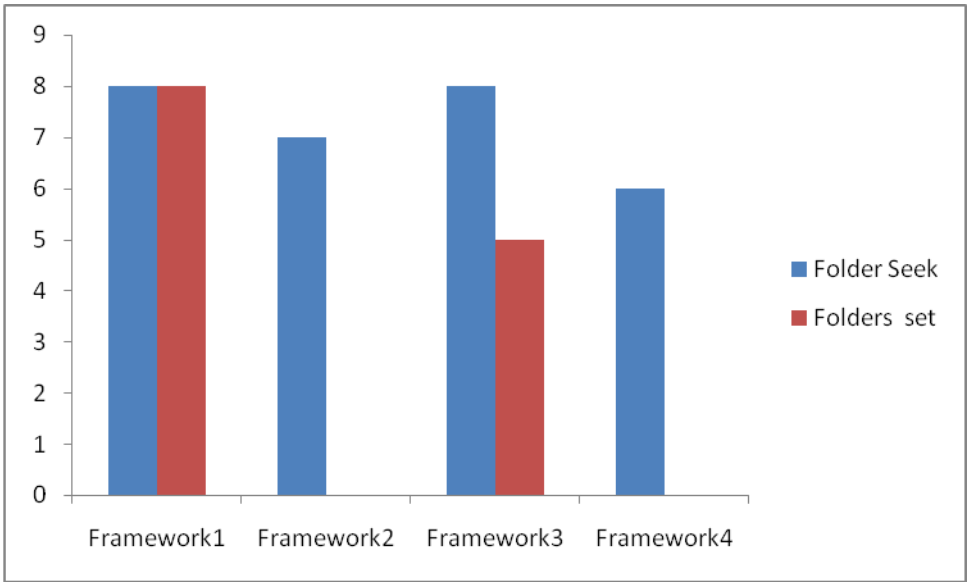

Figure 3 Graph shows number of requested and Allocated files

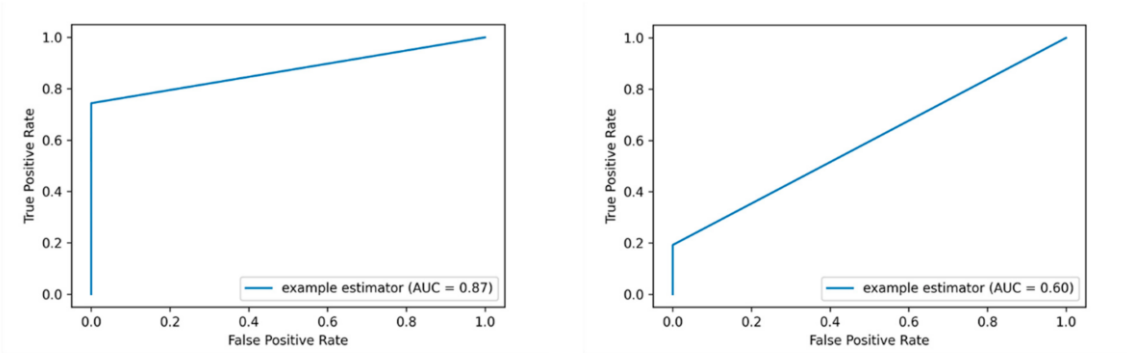

Fig 4 : Graph showing area under ROC curve

The performance evaluation of the algorithm done by analyzing their corresponding ROC curves which is shown in fig 4.(a)and 4.(b). . As seen in Snapshot 4(a), the Data Allocation algorithm-related cure has a higher tendency to 1 (upper left corner), indicating that it has very high-performance statistics as compared to other existing algorithms.

The proposed algorithm's area under the ROC curve (AUC) is 87% which means almost 87 samples out of 100 are transferred correctly. Similarly, for existing leakage detection algorithm is 60%. The variables covered here include true positive rate, false positive rate, and so on. The proposed approach has higher prediction accuracy (lower error rate) than other current algorithms.

V. CONCLUSION

Undetected data leakage is a concern. Your employee, as an insider, may intentionally or accidentally leak critical information. Without your knowledge, this confidential material could be distributed electronically via e-mail, Web sites, FTP, instant messaging, spreadsheets, databases, and any other electronic methods possible. To estimate the danger of spreading data, two factors are necessary. The first is a data allocation strategy that helps to distribute tuples among customers with minimal overlap, and the second is determining guilt likelihood based on the overlap of his data set with the stolen data set.

There is a huge need for protection against data leakage over sensitive data. Watermarking those sensitive data’s for protection may create additional data-stealing issues. This is not a perfect way to prevent data from unknown access and transferred data leaks. The need for new detection and prevention technique may lead to the perfect Data Leakage/Leaker Detection. In the research proposal, the concept is added to detect data leakage and preventing sensitive data’s among other users. This algorithm predicts the future data leakers by analyzing the user data log and behavior. The guilty user-finding phase helps to track all the data leakers. The Data Movement Tracking has developed to identify and prevent the guilty users. The proposed research analyzed the data leaker prediction based on the activity log. There is a chance for low-level accuracy. The issue is extending the allocation strategies such that they can manage user requests in the manner about an online social network. The solutions provided assume that the user has a fixed set of requests that are known in advance.

REFERENCES

[1] Garfinkel, Robert, Ram Gopal, and Paulo Goes. "Privacy protection of binary confidential data against deterministic, stochastic, and insider threat." *Management Science* 48, no. 6 (2002): 749-764.

[2] Kuwatly, Iyad, Malek Sraj, Zaid Al Masri, and Hassan Artail. "A dynamic honeypot design for intrusion detection." In *Pervasive Services, 2004. ICPS 2004. IEEE/ACS International Conference on*, pp. 95-104. IEEE, 2004.

[3] Resmi, A. M., and R. Manicka Chezian. "An extension of intrusion prevention, detection and response system for secure

content delivery networks." In *Advances in Computer Applications (ICACA), IEEE International Conference on*, pp. 144-149. IEEE, 2016.

[4] Lin, Lang, Wayne Burleson, and Christof Paar. "MOLES: malicious off-chip leakage enabled by side-channels." In *Proceedings of the 2009 International Conference on Computer-Aided Design*, pp. 117-122. ACM, 2009.

[5] Shabtai, Asaf, Yuval Elovici, and Lior Rokach. "*A survey of data leakage detection and prevention solutions"*. Springer
Science & Business Media, 2012.

[6] C. Mercy Praba, and Dr.G. Satyavathy, "A Technical Review on Data Leakage Detection and Prevention Approaches." *Journal of Network Communications and Emerging Technologies (JNCET),www.jncet. org* ,Volume 7, Issue 9, September 2017.

[7] Shu, Xiaokui, Jing Zhang, Danfeng Daphne Yao, and Wu-Chun Feng. "Fast detection of transformed data leaks." *IEEE Transactions on Information Forensics and Security* 11, no. 3 (2016): 528-542.

[8] Guevara, César, Matilde Santos, and Victoria López. "Data leakage detection algorithm based on task sequences and probabilities." *Knowledge-Based Systems* 120 (2017): 236-246.

[9] Shu, Xiaokui, Danfeng Yao, and Elisa Bertino. "Privacy-preserving detection of sensitive data exposure." *IEEE transactions on information forensics and security* 10, no. 5 (2015): 1092-1103.

[10] Dr. G. Satayavathy and C. Mercy Praba, "A Study on Cyber Physical System and Network Security." CiiT International Journal of Networking and Communication Engineering, Vol 9, No 3, March 2017, 0974-9713.