Data Leakage Detection and Data Prevention Technique

Mrs.Nisha D.Gaikwad[1], Prof Dr.D S. Bhosle[2]
[1]Department of Computer Science and Engineering,
AMGOI Vathar
[2]Department of Computer Science and Engineering,
AMGOI Vathar

Abstract: Data Leakage is the unplanned or inadvertent appropriation of private or sensitive information to an unapproved subject. Sensitive information in organizations and associations incorporate protected innovation (IP), money related data, understanding data, individual credit card information, and other data relying upon the business. Data Leakage represents a significant issue for organizations as the number of occurrences and the cost to those encountering them keep on increasing. Data Leakage is improved by the way that transmitted information (both inbound and outbound), including messages, texting, site structures, and document exchanges among others, are to a great extent unregulated and unmonitored on their way to their goals. The potential harm and antagonistic outcomes of a data leakage occurrence characterized into two classes as Direct and Indirect Losses. Coordinate misfortunes allude to substantial harm that is anything but confusing to quantify or to gauge quantitatively. Backhanded misfortunes, then again, are considerably harder to evaluate and have a substantially more extensive effect as far as cost, distance, and time. Direct losses include contraventions of regulations (such as protecting privacy information, files data) resulting in fines, settlements or customer enumerations fees litigation involving lawsuits loss of future sales costs of investigation and remedial or restoration fees. Indirect losses include varying share price as a result of severe publicity damage to a company's goodwill and reputation customer abandonment and exposure of intellectual property to competitors.

IndexTerms - *Data leakage detection, privacy information, Deletion, Corrupt data, Encryption, Prevention.*

I. Introduction

Data leakage is a failure condition in information systems in which information is destroyed by failures or neglect in storage, communication, or processing. Information systems implement backup and disaster recovery tools and processes to prevent data loss or restore lost data. Data leakage is distinguished from data unavailability, such as may arise from a network outage. Although the two have substantially similar effects, data unavailability is temporary, while data loss may be permanent. Data leakage is also distinct from data spill, although the term data loss. Data leakage incidents can also be data spill incidents, in case media containing sensitive information is lost and subsequently acquired by another party. However, data leak is possible without the data being lost on the originating side.
There are ten common causes of data loss.

1. **Accidental Deletion of Data:** There are times when accidentally delete a file or a program from your hard drive. This is an unintentional deletion which may go unnoticed for a long time. Administrative errors also fall under this class. The best idea is to think before deleting any data or program.
2. **Incidental drive format**: Users unexpected format their runs and this result in immediate loss of data. However, it is possible to improve your data in a place like this. Get help from experts.
3. **Accidental Damage:** If a drive or disk is harmed or accidentally lost, this may cause trauma and loss of data. Data retrieval is also possible in this case.
4. **Natural Disaster:** Your hard drive can destroy due to fire, flood or some other surprising disasters.
5. **Useful Deletion of Data:** You may have deleted a file
deliberately from your system and later decided you want the file back to your system. You can still recover your data from the recycle bin. If you have left your recycle bin, you can use software recover deleted recycle bin files.
6. **Power Failure:** If you undergo power failure before you can save your work, you may lose valuable data. The advice is to keep collecting your work.
7. **Corrupt Data:** If your file system or database is corrupt, then you are bound to lose data. Again it is likely to recover data from a corrupt file system with the right software tool.
8. **Software Failure:** When utilization software suddenly crashes or freezes while working, this may result in severe damage to the hard drive. This causes the program to close suddenly, and all unsaved work is lost.
9. **Virus Attack:** If a machine is deeply infected by viruses and worms, spyware, adware and some deadly computer dependents, this can be very deadly, and it may result to total corruption and loss of data. Installing a good anti-virus will reduce the possibility of having a fatal virus attack.
10. **Spiteful Attack:** Professional hackers can invade your machine and damage your system. This will undoubtedly lead to the loss of data.

II. Related Work

The guilt detection approach is related to the data provenance problem [4] tracing the linkage of S objects implies the detection of the guilty agents inherently. Tutorial [5] provides a good overview of the research conducted in this field. Suggested solutions are domain-specific, such as linkage tracing for data warehouses [6], and assume some prior knowledge on the way a data view is created out of data sources. Our problem formulation with objects and sets is more general and simplifies lineage tracing, As far as the data allocation strategies are concerned, work is most related to watermarking which is used for establishing original ownership of allocated objects. [9], and audio data [7] whose digital representation includes significant redundancy. Recently, [2], [11], [8], and other work has studied for marks insertion into relational data.

Data Leakage Detection:

With the fastest growth in the database **of** business on the internet, the data may be unsafe after passing through the unsecured network. The data buyers may hesitate to buy the data service for the following suspicion. First, the data receiver may suspect that the data are tampered with by an unauthorized person. Second, they may suspect the data received are not producing and provided by the authorized suppliers. Third, the suppliers and purchasers actually with different interest should have different roles of rights in the database management or use.

In section II a guilty agent is introduced which is developed for assessing the "guilt" of agents and also present algorithms for distributing objects to agents, Sections III and IV, present a model for calculating "guilt" probabilities in cases of data leakage. Finally, in Section V, evaluating the strategies in different data leakage scenarios, and check whether they indeed help to identify a leaker.

III. Challenges

a) Encryption: For preventing data leaks in transit are hindered due to encryption and the high volume of electronic communications. While encryption provisions mean to assure the confidentiality, authenticity, and integrity of the data, it also makes it hard to identify the data leaks are happening over encrypted channels. Encrypted emails and file transfer protocols such as SFTP imply that complementary DLP mechanisms should be employed for more excellent coverage of exposure channels. Employing data leak prevention at the endpoint – outside the encrypted channel has the potential to identify the leaks before the communication is encrypted**.**

b) Access Control: It provides the first line of defense in DLP. However, it does not have the proper level of granularity and may be outdated.

c) Semantic Gap in DLP: DLP is a multifaceted problem. The sense of a data leak is likely to vary between organizations depending on the sensitive data to be preserved, the degree of interaction between the users and the available communication channels. The current state-of-the-art mainly concentrates on the use of abuse detection and post-mortem investigation (forensics). The common shortcoming of such methods is that they lack the explanation of the events being monitored. When a data leak is defined by the communicating parties as well as the data exchanged during the communication, a simple pattern matching or access control design cannot infer the nature of the communication. Therefore, data leak prevention mechanisms need to keep track of who, what and where to be able to defend against complex data leak scenarios. The classification by leakage flow is essential to know how the events may be restricted in the future and can be classified as physical or logical.

In data leak detection model, two types of sequences analyzed: sensitive data sequence and content sequence.

- The content sequence is the sequence to be examined for leaks. The content may be data extracted from file systems on personal computers, workstations, and servers or payloads extracted from supervised network channels(details are discussed below).
- Sensitive data sequence contains the information (e.g., Customers' records, proprietary documents) that need to be protected and cannot be exposed to illegal parties. The sensitive data flows are known to the analysis system.

Work concentrating on detecting inadvertent data leaks assuming the content in the file system or network traffic(over supervised network channels) is available to the inspection system. A supervised network channel could be an unencrypted channel or an encrypted channel where the content in it can be extracted and checked by an authority. Such a channel is widely used for advanced NIDS where MITM (man-in-the-middle) SSL sessions are established instead of ordinary SSL sessions. Preventing intentional or malicious data leak, especially encrypted leaks, requires different approaches and remains an active research problem. In the current security model, assumes that the analysis system is secure and trustworthy. Privacy-preserving data-leak detection can be achieved by leveraging particular protocols.

IV. Objectives

The different objectives of the proposed system are

- To define and prepare the dataset for the proposed system.
- Design data-movement tracking approaches for data leak prevention on a host.
- Use of parallel versions prototype to provide substantial speedup and high scalability.
- The design approach to detecting leaks of various sizes, ranging from tens of bytes to megabytes.
- Design and implement Data Leakage prevention mechanism.

Traditionally, leakage detection is handled by watermarking, e.g., a unique code is set in each distributed copy. If that copy is later found in the hands of an unauthorized party, the leaker can be recognized.

Guilty Agent

To detect when the distributor's sensible data has been leaked by agents, and if possible to identify the agent that leaked the data. Dis tress is an advantageous technique where the data is modified and made "less sensitive" before being handed to agents.

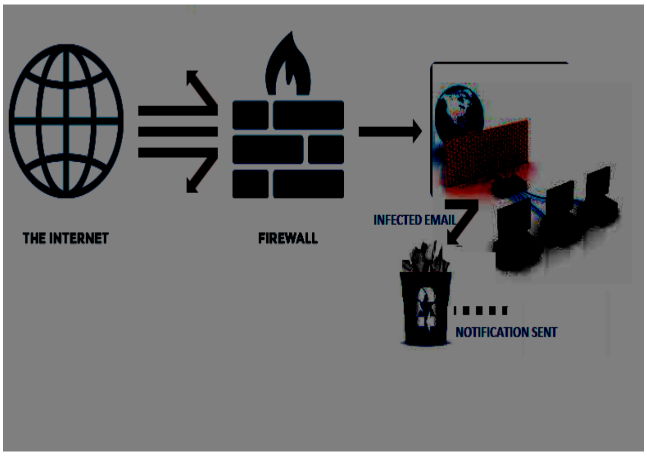

Figure 1.Data Leakage Detection

In the figure 1 general architecture of data leakage detection is shown and it is used to understanding basic flow of the system.

V. PROPOSED SYSTEM

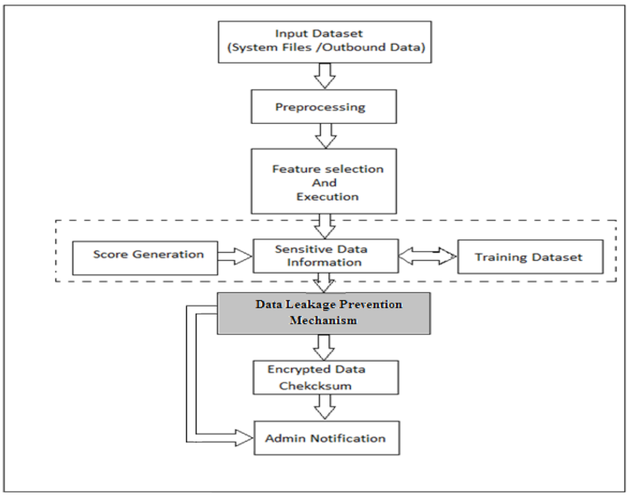

Figure 2 proposed system architecture

The proposed system is shown in figure.2 The working of the proposed system is described with the following blocks.

The working of the proposed system is described with the following blocks.

Module 1: Input Dataset: -

Using user interface tool select input as system files and outbound data for detecting inadvertent data leaks. The content sequence is the sequence to be examined for leaks.

Module 2: Preprocessing: -

Outbound data is used after preprocessing. The task preprocessing is used to change data into tool readable format.

Module 3: Feature selection and Exaction: -

Features are selected from the file system to extract the information and to detect data leaks from outbound data. The content may be data extracted from filesystems on workstations and servers or payloads extracted from supervised network channels

Module 4: Training Data: -

Training is provided to classify the input to further processing. as per the training, data will classify into different categories, i.e., Normal and Abnormal.

The content and the sensitive data sequences both are known to the analysis system. A data loss is detected when the detection system finds a piece of sensitive data in the content sequence (Fig. 1), but the appearance is not allowed in the sensitive data storage and sharing policy. The two assumptions can be removed when proposed system alignment is performed utilizing secure multiparty computation or other privacy-preserving techniques. Not aim at detecting stealthy data leaks that an attacker encrypts the sensitive data by herself before leaking it.

Work presents an efficient sequence comparison technique needed for analyzing a significant amount of content for sensitive data exposure. Work-flow in Fig. 1 is a detection approach consists of a special sampling algorithm and a corresponding alignment algorithm working on preprocessed n-grams of sequences. The pair of algorithms computes a quantitative similarity score between sensitive data and content. Local alignment, as opposed to global alignment, is used to identify similar sequence segments, enabling the detection of partial data leaks.

The workflow includes Extraction, Preprocessing, Sampling, Alignment, and Decision operations. The Extraction operation collects content. The Preprocessing operation prepares the sequences of n-grams for both the content and sensitive data. The Sampling operation generates samples from both sensitive data and content sequences. The Alignment operation performs local alignment between the two sampled sequences to compute their similarity. Finally, the Decision operation consumes and reports leaks according to the sensitive data sharing policy.

VI. Technical Challenges

1) High Detection Specificity:

In the data-leak detection model, high specificity refers to the ability to distinguish correct leaks from coincidental matches. Coincidental matches are false positives, which may lead to false alarms. Existing set-based detection is order-less, where the order of matched shingles (n-grams) is ignored. Order-less detection may generate coincidental matches, and thus be having a lower accuracy of the detection. In comparison, the proposed system alignment-based method has high specificity.

2) Pervasive and Localized Modification: Sensitive data could be modified before it is leaked out. The modification can occur throughout a sequence (pervasive modification). The modification can also only affect a local region (local modification). Here describe some modification examples:

- Character replacement, e.g., WordPress replaces every space character with a + in HTTP POST requests.
- String insertion, e.g., HTML tags inserted throughout a document for formatting or embedding objects.
- Data truncation or partial data leak, e.g., one page of a two-page sensitive document is transmitted.

Problem Formulation

To design and implement sequence alignment techniques for detecting complex data-leak patterns. The approach is intended for detecting inexact sensitive data patterns. This detection is combined with a practically identical inspecting calculation, which permits one to think about the comparability of two independently sampled sequences. A designing system to achieves good detection accuracy in recognizing transformed leaks. Parallelized adaptation is used for calculations for preparing high examination throughput.

Sensitive Data Analysis

Content-centric data protection technologies such as DLP rely heavily on the proper classification of sensitive information. DLP policies are defined to target sensitive documents and their handling within an organization primarily. Streamlining sensitive data handling applications from creation to archiving and deletion through policies and practices should be a crucial step for successful DLP enforcement. The identification and classification of sensitive data according to the policies and guidelines of the organization are essential steps for achieving a comprehensive data protection strategy.

Policy Request Management

Creating relevant and meaningful policies is central to the DLP strategy. Depicts typical DLP operational activities in an organization. Policies are created to monitor or block (prevent) sensitive data from leaving an organization's network. A structured policy request and review process can help to ensure that the policies defined are essential and relevant as well as do not overlap with existing policies. Policy changes or modifications should handle through a controlled process. DLP policies also need a periodic review to adapt to changing technologies, business practices, and new risk scenarios.

Implement useful event review and investigation mechanisms

Events triggered by policy violations and the resulting activity logs (when blocking or monitoring) are critical outputs from a DLP tool that provide valuable information and insight. An active and warm review mechanism is required to realize the benefits of the solution.

Provide analysis and meaningful reporting

Events triggered by DLP policies provide useful insight on where, when and how the sensitive data are stored and handled within the organization. Events can investigate by breaking them down into individual policies, departments, regions, and trends. The aggregate picture could provide insights into current data-handling practices and where the industry requires additional awareness and training.

Implement security and compliance measures

A DLP system collects a significant amount of data, some of which may be personal. The administration of personal data collected should comply with data privacy laws and regulations of the countries from which the data are handled. The data can also be business delicate; therefore, it is critical to managing the DLP system and the data captured securely and in compliance with applicable laws and regulations. As with other technologies, DLP has its flaws in preventing or detecting every data loss event in a dynamic technology world.

VII. Implement an organizational data flow and oversight mechanism

Implementation Steps

Potential data leakage can manage by different data loss tools, also known as data leakage blocking or content monitoring and filtering tools. It is accomplished through identifying content, tracking activity and potentially blocking sensitive data from being moved.

Data Leakage prevention can be managed through the following steps

1. By performing content-aware, deep packet inspection on the network traffic as well as email and various other protocols. Content-aware data leakage prevention identifies critical data based on policies and rules previously determined and set up. It can be deployed at different stages: they are on the network, Endpoint or stored data.
2. By ensuring that complete sessions are always being tracked for analysis and not only single packets.
3. By using both statistical and linguistic techniques for analysis, like expressions, document fingerprinting and machine learning.
4. By detecting, blocking and controlling the use of specific content based on rules and policies, thus not allowing saving, printing, and forwarding of specific predetermined content.
5. By monitoring network traffic, email traffic and multiple channels through one product and an individual management interface
6. By blocking policy violations over email and other external communication methods like IM.
7. By ensuring an end user policy compliance solution, by controlling what end-users do on their computers through managing the use of connected devices and network interfaces, managing the applications they use and by managing websites which users can access. An endpoint solution manages the threat of portable storage devices by giving administrators control over what devices are in use when they are in use and by whom as well as knowledge of the data that has been copied. The activity

of media players, USB drives, memory cards, PDA's, mobile phones, network cards, etc. can be logged, as well as centrally disabled if need be

8. By encrypting all communications and data (email, file shares, hard drives, external storage and removable media)
For more details steps to data leakage prevention – by Tech Genix. http://techgenix.com/ Data-Leakage-Prevention/ [26].

VIII. Experimental Result

We conducted experiments with simulated data leakage problem to evaluate our allocation algorithms. We implemented the leakage detection algorithms in JSP. JSP is intended to be a simple, modern, general-purpose, should provide support for software engineering principles Software robustness, persistence, and programmer productivity are essential. The language is designed for use in developing software components suitable for deployment in distributed environments

The experimental section is divided into two subsections. In the first subsection, we present the algorithm evaluation for the explicit request, and in the second section, we present algorithm evaluation for sample request.

1. Explicit Data Request

With a few things that are shared among multiple agents. These are the most entertaining scenarios since object sharing makes it difficult to distinguish a guilty from non-guilty agents. Scenarios with more objects to distribute and scenarios with objects shared among fewer agents are to handle.

In our scenarios, we considered a set of |T|=500 objects and online requests of agents received one by one with the same condition. Suppose eight objects satisfy this condition so that all the agents will get the same eight tuples. Here in A will be 0 and it is tough to detect a guilty agent in case leakage of any tuples out of these 8. As we get min A > 0 it becomes effortless to find out the guilty agent.

The value of min A will be increased by the addition of fake tuples to the set of agent's request. The fake and original tuples are selected randomly using the e-random algorithm. By allocating 30% fake objects, the distributor can detect a guilty agent even in the worst case leakage scenario, while without fake tuples he will be unsuccessful not only in the worst case but also in the average case.

Table 1 Sample request (Explicit) from five Agents

Agent Name	No_of_Fake tuple added (in %)	The probability of Leakage(in %)
NISHA	150	0.8
MANJIT	150	0.9
VIKRAM	150	0.0
DHANRAJ	150	0.3
SNEHA	150	0.8

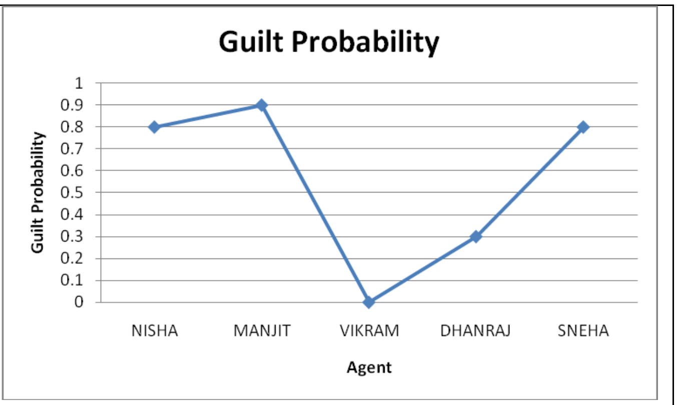

Figure 3. Dependence of guilt probability on the number of fake tuples

The above graph shows that as the addition of fake tuples is constant to the agent's dataset the probability of guilt detection is also increased. But as we discussed previously, we can insert a large number of fake tuples. This may bias the results of agent's survey on that dataset.

2. Sample Data Request

With the sample data request agents are interested in particular objects. Hence, object distribution is not explicitly defined by their requests. The distributor is "forced" to allocate specific objects to multiple agents only if number of requested objects £f=1 exceeds the number of objects in set T. The more data objects agents request in total, the more recipients on average an object has; and more objects are shared among various agents, the more difficult it is to detect a guilty agent.

In our experimental scenario, set *T* has 500 objects, and m_i is the number of tuples given to every agent.
1. Initially take the sample requests from 5 agents. The graph shows that as the overlap of the agent's request with the leaked dataset increases the guilt probability increases.

Table 2 Sample Request from five Agents

Agent Name	Guilt Probability	Overlap with Leaked set %
NISHA	0.8	392/439=89
MANJIT	0.9	428/439=97
VIKRAM	0.0	402/439=91
DHANRAJ	0.3	428/439=97
SNEHA	0.8	402/439=91

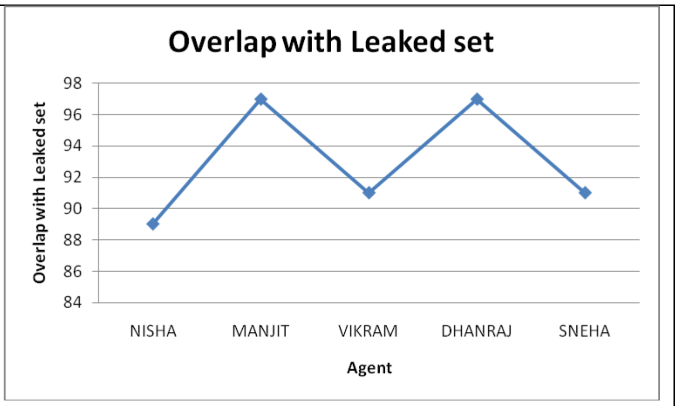

Figure 4. Guilt probabilities for sample request (Implicit) from ten agents

IX. Conclusion

In the data leakage, we can identify and block the data from the leak by using some algorithms and methods. In a world, there would be no need to hand over sensitive data to agents that may unknowingly or maliciously leak it. Moreover, even if we had to hand over sensible data, in a perfect world, we could watermark each object so that we could investigate its beginnings with absolute certainty.

Whenever valuable or sensitive business information such as customer or patient data, source code or design specifications, intellectual property and trade secrets are handed over to supposedly trusted third parties, then there is the possibility of some data leaked out. When such a leader leaves the company then sometimes it makes the unprotected and goes outside the jurisdiction. This uncontrollable data leakage put the business in a vulnerable position. Once this data is no longer inside the domain, the company is at serious risk. When cybercriminals "cash out" or sell this data for profit it costs our organization money, damages the competitive advantage, brand, and reputation and destroys customer trust. In many situations, we must indeed work with agents that may not be 100% trusted, and we may not be defined if a leaked thing came from an agent or another source, since certain data cannot admit watermark. My presented model assesses the “guilt” of agents. The main focus of this scheme is the data allocation problem. It specifies how the distributor can "intelligently" give data to agents to improve the chances of detecting a guilty agent. By adding fake objects to the distributed set, the distributor can find the guilt agent easily.

References

[1] A. Bonaccorsi, “On the Relationship between Firm Size and Export Intensity,” Journal of International Business Studies, XXIII (4), pp. 605-635, 1992.

[2] A. Bonaccorsi, “On the Relationship between Firm Size and Export Intensity,” Journal of International Business Studies, XXIII (4), pp. 605-635, 1992. (journal style)

[3] Panagiotis Papadimitriou and Hector Garcia-Molina, “Data Leakage Detection,” IEEE Trans, Knowledge and Data Engineering, vol. 23, no. 1, January 2011.

[4] 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.

[5] 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.

[6] 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.

[7] P. Buneman and W.-C. Tan,“Provenance in Databases,” Proc. ACM SIGMOD, pp. 1171-1173, 2007.

[8] Y. Cui and J. Widom, “Lineage Tracing for General Data Warehouse Transformations,” The VLDB J., vol. 12, pp. 41-58, 2003.

[9] S. Czerwinski, R. Fromm, and T. Hodes, “Digital Music Distribution and Audio Watermarking,”http://www.scientificcommons.org/430256 58, 2007.

[10] 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.

[11] F. Hartung and B. Girod, “Watermarking of Uncompressed and Compressed Video,” Signal Processing, vol. 66, no. 3, pp. 283-301, 1998.

[12] 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.

[13] 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.

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

[15] R. Sion, M. Atallah, and S. Prabhakar, “Rights Protection for Relational Data,” Proc. ACM SIGMOD, pp. 98-109, 2003.

[16] XiaokuiShu, Jing Zhang, Danfeng (Daphne) Yao,"Fast Detection of Transformed Data Leak," IEEE transactions on information forensics and security, vol. 11, no. 3, March 2016

[17] Shapira, Yuri, Bracha Shapira, and Asaf Shabtai. "Content-based data leakage detection using extended fingerprinting." arXivpreprint arXiv:1302.2028 (2013).

[18] Alneyadi, Sultan, ElankayerSithirasenan, and VallipuramMuthukkumarasamy."Detecting Data Semantic: A Data leakage prevention Approach."Trustco/BigDataSE/ISPA, IEEE.Vol. 1.IEEE, 2015.

[19] Wen, Yan, Jinjing Zhao, and Hua Chen."Towards Thwarting Data Leakage with Memory Page Access Interception."Dependable, Autonomic and Secure Computing (DASC), IEEE 12th International Conference on. IEEE, 2014.

[20] Alneyadi, Sultan, ElankayerSithirasenan, and VallipuramMuthukkumarasamy."Adaptable n-gram classification model for data leakage prevention."Signal Processing and Communication Systems (ICSPCS), 7th International Conference on.IEEE, 2013.

[21] Shu, Xiaokui, and Danfeng Daphne Yao. "Data leak detection as a service."Security and Privacy in Communication Networks.Springer Berlin Heidelberg, 2012.222-240.

[22] Wu, Jiangjiang, et al. "An active data leakage prevention model for insider threat." Intelligence Information Processing and TrustedComputing (IPTC), 2011 2nd International Symposium on. IEEE, 2011.

[23] Papadimitriou, Panagiotis, and Hector Garcia-Molina."A model for data leakage detection."Data Engineering, ICDE'09.IEEE 25th International Conference on.IEEE, 2009.

[24] Marecki, Janusz, MudhakarSrivatsa, and PradeepVarakantham."A Decision-Theoretic Approach to Data leakage prevention."Social Computing (SocialCom), IEEE Second International Conference on. IEEE, 2010.

[25] Shaj.v ,K.P. Kaliyamurthie “A Review on Data Leakage Detection” International Journal of Computer Science and Mobile Computing IJCSMC, Vol. 2, Issue. 4, April 2013, pg.577 – 581

[26] http://techgenix.com/Data-Leakage-Prevention/