International Journal of Future Innovative Science and Engineering Research (IJFISER)
Volume - 2, Issue - II ISSN (Online): 2454- 1966

Research Manuscript Title

PRIVACY PRESERVING DATA LEAK DETECTION IN LARGE SCALE ORGANIZATIONS

Hemalatha.N.C, Somasundaram.R, Mythili Thirugnanam

P.G Student,CSE Department, Assistant Professor,CSE Department, Associate Professor, SCOPE, Arulmigu Meenakshi Amman College Of Engineering, VIT University,Vellore – 632014,

E-Mail: hema_sunrose@yahoo.co.in , somsb88@gmail.com , tmythili@vit.ac.in

JUNE – 2016

www.istpublications.com

Privacy Preserving Data Leak Detection in Large Scale Organizations

Hemalatha.N.C, Somasundaram.R, Mythili Thirugnanam

P.G Student,CSE Department, Assistant Professor,CSE Department, Associate Professor, SCOPE,
Arulmigu Meenakshi Amman College Of Engineering, VIT University,Vellore – 632014,

E-Mail: hema_sunrose@yahoo.co.in , somsb88@gmail.com , tmythili@vit.ac.in

ABSTRACT

Sensitive data disclosure is a leading cause of data exploitation world-wide. The administrators are imposed with a task of protecting confidential information from leaving their networks. Among different accidental secured information disclosure cases. Human mistakes are one of the fundamental reasons of information disclosure. In existing works the realizations of data-leak detection require the plaintext sensitive data. However, this requirement is undesirable, as it may threaten the confidentiality of the sensitive information. In furtherance to address this problem, in this paper, a host-assisted mechanism for complete data leak prevention as well as data leak detection (DLD) in large scale organizations is designed. The system checks for ongoing traffic for known confidential information and any oppositions from leaking such information even in any altered form using algorithm such as Code texted Levenshtein algorithm and with efficient use of available resources. For appropriate detection, the algorithm relies on calculating the maximum number of possible alterations even in the encoded form. Evaluation results under various data-leak scenarios and setups shows that the method can support accurate detection with very small number of incorrect alarms, even when the presentation of the data has been altered. It also indicates that the detection accuracy does not degrade.

Keywords—Sensitive Data Leak Detection, Code texted Levenshtein technique, network security, privacy**.**

I. INTRODUCTION

Today's individual computational gadgets, from desktops to versatile PCs, advanced cells and purchaser hardware, run a wide assortment of utilizations that send client data over the system to different gatherings. This data is utilized as a part in numerous profitable ways e.g., current area data is utilized to interface with maps [3].In any case, breaks of individual data are additionally a potential reason for concern since they might attack security in ways that were not expected or coveted by clients, e.g., when outsiders assemble definite profiles of client conduct. So, the proprietors of data must know about the data uncovered by their applications or by their clients with the goal that they can evaluate whether it causes a risk to their protection and classification issues [6]. Likewise, individuals who direct machines stand to profit by learning of what applications are uncovering which data to whom, with the goal that they can better survey and set protection and security arrangements. Be that as it may, it is extremely hard to know how individual data is uncovered by arranged applications. Clients should depend vigorously on portrayals gave by application engineers (or advertisers) since there is no genuine, autonomous approach to naturally confirm what is revealed by whom. Apparatuses to check system movement for individual data are ordinarily constrained to a little arrangement of very much characterized data, for example, MasterCard or government managed savings numbers [21]. Along these lines, numerous data holes are found unintentionally. So to achieve protection and security issues of an association a framework must be utilized at the level of association itself for forestalling and identifying potential holes of touchy data by savvy strategies[12].

Run of the mill ways to deal with forestalling information break are under two classifications

1. Host-based arrangements. 2. System based arrangements.

Host-based methodologies might incorporate (i) encrypting information very still when not utilized [2], (ii) recognizing stealthy malware with antivirus filtering or observing the host [10, 11], and (iii) upholding strategies to limit the exchange of sensitive information to guarantee protection of information proprietors data [5].

These information are motivated to make study about difficulties in sensitive data disclosure and detection methods.

II. Related Work

There have been many works done in sensitive data leak detection. In the large organizations at the point when connected to genuine web searching activity, the calculations could markdown 98.5% of measured bytes and viably disengage data spills. In any case, Instead of attempting to detect the presence of sensitive data these calculations just evaluates data leaks limit in system movement and their objective is to quantify and compel its maximum volume information leaks in system activity [2].

Tracking of data from its creation (origin) to its current state or its end state will enable the full transparency and accountability in cloud computing environments. Techniques for tracking end-to-end data provenance, a meta-data describing the derivation history of data was proposed earlier [18]. This breakthrough is crucial as it enhances trust and security for complex computer systems and communication networks. By analyzing and utilizing provenance, it is possible to detect various data leakage threats and alert data administrators and owners; thereby addressing the increasing needs of trust and security for customers' data. They also present a rule-based data provenance tracing algorithms, which trace data provenance to detect actual operations that have been performed on files, especially those under the threat of leaking customer's data.

The algorithms were able to (1) Detect file copying, renaming and movements between directories within a local machine in the cloud, and (2) Detect the file sending and receiving across different machines in the same cloud, and even email client file leakages. But these algorithms does not address real issues because the algorithms were (1) Unable to detect the state where a new file is created (2) Unable to distinguish the state where a process reads a file and creates another file [18].

Another proposes a new automatic approach that applies Named Entity recognition (NER) to prevent data leaks. They conduct an empirical study with real-world data and show that this NER-based approach can enhance the prevention of data losses. In addition, they created a prototype able to alert users about possible data leaks [11].But the use Named Entity Recognition (NER) is only for natural languages not for encrypted information.

In contrast with host-based methodologies, system construct information spill discovery centers in light of dissecting the (decoded) substance of outbound system packets for delicate data. For instance, a notable arrangement requires assessing each packet for the event of any of the delicate information characterized in the database [2]. Such arrangements create alarms if the sensitive information is found in the active movement. Be that as it may, this guileless arrangement requires putting away such sensitive information in plain-text at the system interface, which is very undesirable [23]. Another inspiration for our protection safeguarding DLD work is distributed computing, which gives a characteristic stage to directing information spill discovery by cloud suppliers as an extra administration.

In distributed computing situations, an organization (Data owner) might have as of now outsourced its administrations to a cloud supplier, for example, the email administration for its own particular workers [17].

The cloud supplier might offer extra administrations, for example, investigating email movement for accidental information leak and serves as a DLD supplier. This extra DLD administration requires insignificant changes to the cloud supplier's base and makes the cloud benefit more appealing. Be that as it may, protection is a noteworthy barricade for acknowledging outsourced information spill recognition.

Routine arrangements require the information proprietor to uncover its delicate information to the DLD supplier. In any case, the DLD supplier is constantly displayed as a genuine however inquisitive (otherwise known as semi-fair) enemy why should trusted perform the review, yet might endeavor to find out about the information. Because of the risk of system

based data spills, specialists have created information misfortune counteractive action (DLP) Systems. These DLP frameworks work via looking through outbound system activity for known delicate data. In this manner, inside of an organization the work of DLD supplier should be possible by the DLP frameworks that demonstrations like a firewall examining every last information outsourced by the representatives of the association without their insight into being reviewed. There are a few practical sense of DLD arrangement as takes after.

There are two specialized difficulties connected with system based DLD location. To begin with, the DLD supplier picks up learning about the sensitive information when the activity contains a break. The test is the means by which to confine the level of data that can be learned by the DLD supplier in the event of information holes – the DLD supplier has the entrance to the plain-text payload [23]. The second test is the way to make the location clamor tolerant, for instance, the captured payload might contain random bytes or the touchy information is truncated. The answers for these difficulties are to deliberately randomize the discovery for enhancing the protection, and select nearby elements to accomplish the clamor resilience [7]. Subsequently a protection model indicating the foe's abilities and suspicions is given. Most of the works dealt with only preventing the data in the natural form but not in the altered form .based on the study Preventing sensitive data from being compromised is an important and practical research problem. The privacy goal is quantified and restricting the probability that the DLD provider identifies the exact value of the sensitive data even in altered form.

In this proposed work, a system that finds data leaks of sensitive information in transmissions, without the knowledge of data owner, by unauthorized persons is developed. The system discovers leaks even when the structure of information being leaked was previously unknown modified form, and does so without requiring a deep or intrusive instrumentation of the computing system and in a cost effective manner.

III. PROPOSED WORK

Sensitive information might be spilled for a few reasons. However the point is to identify coincidental information spill in our threat model.

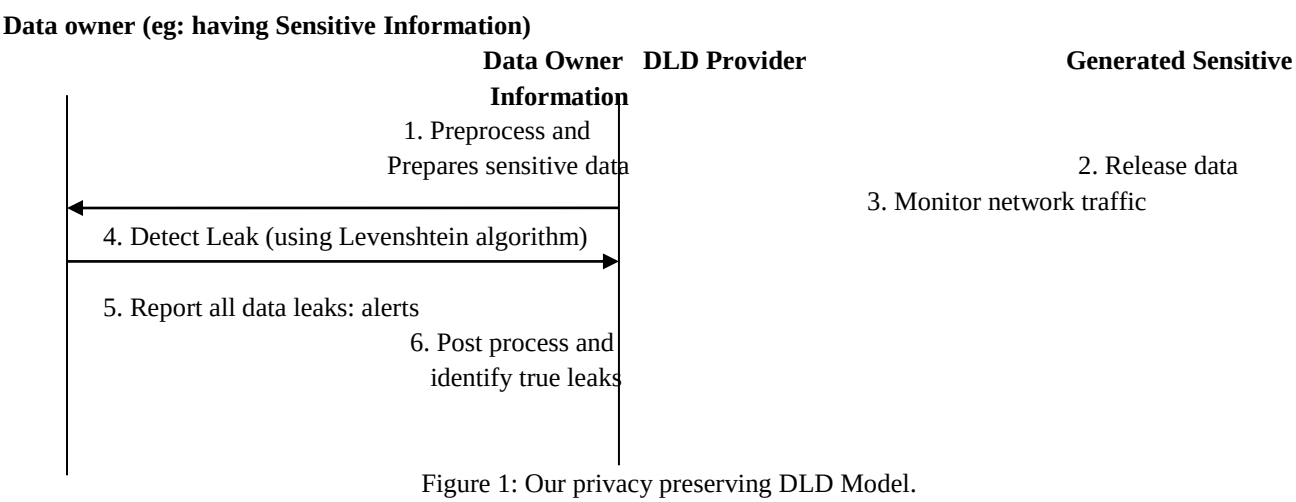

Figure 1: Our privacy preserving DLD Model.

Case I Inadvertent information spillage [17]: The delicate information is coincidentally spilled in the outbound activity by a true client. The methodology concentrates on recognizing this kind of unplanned information spills over the system. Incidental information breaks might happen in various courses, e.g., because of human mistakes, for example, neglecting to utilize encryption, imprudently sending an inner email and connections to outsiders, or because of utilization faults. Case II Malicious information spillage [10]: A maverick insider or noxious and stealthy programming might take delicate individual or authoritative information from a host. Case III Legitimate and expected information exchange [18]: The delicate information is sent by a real client proposed for true purposes. There are two players in this model: the Organization (i.e.,

Data owner or information proprietor) and the Data Leakage Detection (DLD) Provider, who may be a third-party service provider or the data owner itself. Organization possesses the delicate information and approves the DLD supplier to review the system movement from the authoritative systems for oddities, in particular coincidental information spill. In any case, the association would not like to straightforwardly uncover the delicate information to the DLD supplier. DLD supplier assesses the system movement for potential information spills. The examination can be performed disconnected from the net likewise without bringing on any ongoing real time delay or postponement in directing the packets.

IV. IMPLEMENTATION MODEL

The workflow in a network-based data-leak detection framework [21] is as follows: Data pre-processing by the data owner, Traffic pre-processing and detection by the DLD provider (which may be a specialized system or a third party service provider), and analysis by the data owner [28]. Data pre-processing is where the data owner takes the sensitive data-set and computes the corresponding set of digests. Traffic pre-processing and detection is where the DLD provider gathers network packets and inspects the content for data-leaks. Analysis is where the data owner efficiently examines the alerts generated by the DLD provider, identifies and investigates the true leak instances and ignores false positives.

Data- leakage detection in an organization can be implemented as following steps.

- Generating Sensitive Data
- Content Outsourcing
- DLD Checker
- Sensitive Data Detection

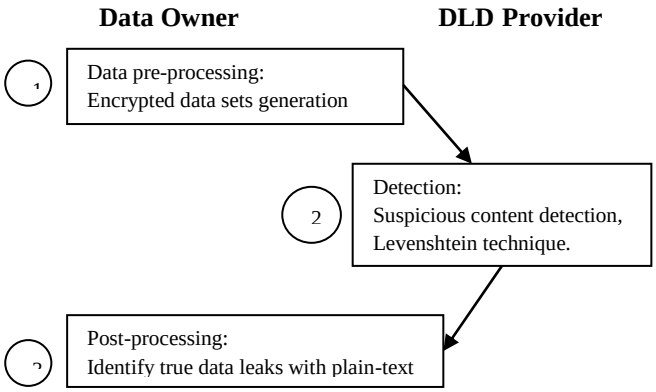

Figure 2: Workload Distribution for DLD Provider and Data Owner.

A. Generating Sensitive Data

The data owner generates a sensitive data and stores it in the database as a dictionary of words (using multi-part file splitter technique). The major task of the File splitter is to provide the user the flexibility of passing the information for implementing the encoding standards as per the specification and algorithms proposed and store the information in a form that is unreadable. The algorithms used in application should confirm the standards of authentication and authorization policies and standards of the organization.

The data owner database contains much sensitive information about many of the file content. So the data in the database must be stored in any human non-readable form. This sensitive information is maintained by Data Leak Detector which can be an implemented as a system or a router or a firewall. Using this only DLD perform data leak detection mechanism.

B. Content Outsourcing

File content or any text format is outsourced from one organization to another organization by the employees of organization. Outsourcing of content is performed by user. The content can be of any file (text, document). Every outsourcing will reach DLD and is transmitted outside the organization premises only if the DLD provider ensures that, the content is being transferred by authorized person(if it is a sensitive data) or the content does not contain any sensitive data even in an modified form. This ensures that only the authorized persons are involved in communication and any attempt made by unauthorized persons to transmit data will be prevented by DLD technique. Here outsourcing is not performed normally, all content will be encoded and then only it transferred. An algorithm such as Base64 can be used for this purpose.

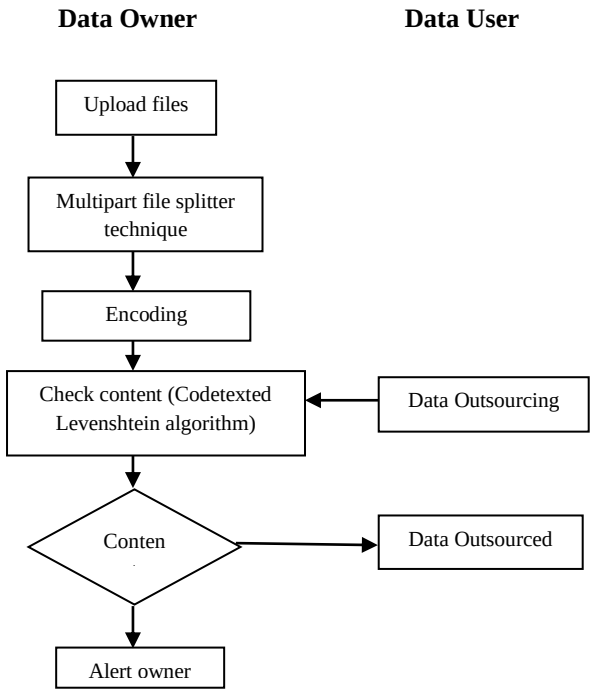

Figure 3: Architectural Overview of Proposed model.

i. **Generating Code texting Technique**

Base64 encoding is sensibly productive and has the upside of being very perfect with most working frameworks since it just uses a restricted character set in its encoded structure.

This calculation is depicted as a feature of the MIME detail. Likewise, RFC 3548 portrays Base64 encoding, however takes a somewhat diverse state of mind to line breaks.Base64 encodes the information three bytes at once. Every piece of three data bytes is encoded to make a square of four printable characters. By detail the encoded information have a CRLF character pair embedded after each 76 characters (or less) of encoded information.

Base64 requires 64 ASCII characters to encode information. This arrangement of characters is called letter set. There are just 62 alphanumeric characters, so Base64 fundamentally utilizes two accentuation characters as a part of expansion to the alphanumeric characters. The characters ""+"" and "/"" were picked (quite a while back) for most extreme similarity with obsolete, non-ASCII character encodings, for example, EBCDIC.

It is hard to utilize these characters is Base64 information in a filename or URL, these characters are dangerous - they are now utilized by some recording frameworks. So an option letters in order is utilized. The Filename Safe Alphabet, which employs:

- "- " (minus) rather than "+" for character 62.
- "_" (underscore) rather than "/" for character 63. Aside

From that the calculation is indistinguishable. The first information can be any length, not as a matter of course a numerous of 3. This implies the last square of double information could be 1, 2 or 3 bytes in length. To code the last square, zeros are added to the last piece to make it a different of 3, and believer it to 4 characters.

C. DLD Checker

Code texted Levenshtein algorithm calculates the least number of edit operations that are necessary to modify one string to obtain another string. The Code texted Levenshtein algorithm distance between two words is the minimum number of single-character edits (i.e. attachments, removals or changes) required to change one word into the other. This can be implemented as modern dynamic programming approach. A two-dimensional matrix, m [0...| X1|, 0...| X2|] is used to hold the edit distance encoded values:

m [imp] = d(X1 [1...i], X2 [1...j])

m [0, 0] = 0
m[i,0] = i, i=1..| X1|
m[0,j] = j, j=1..| X2|

m[i,j] = min(m[i-1,j-1]
+ if X1 [i]= X2 [j] then 0 else 1 if,
m[i-1, j] + 1,
m[i, j-1] + 1), i=1..| X1|, j=1..| X2|

m[,] can be computed row by row. Row m[i,] depends only on row m[i-1,]. The time complexity of this algorithm is O (|X1|*| X2|). If s1 and X2 have a `similar' length, about `n' say, this complexity is O (n2), much better than exponential.

A matrix (m, n) cell is used and the Code texted Levenshtein algorithm distance between m-character prefix of one with n-prefix of other word. The weight or cost of each modification is calculated .Once the threshold value is met it ensures leakage of sensitive information and the owner is alerted.

DLD is the one that checks all the outsourcing content before it transmitted outside the organization's network. All the outsourced contents are checked with sensitive data. All the sensitive data are maintaining index file. From this index file only DLD identify the sensitive data. If the data is undergone any modifications it is essential to identify such data leaks also. So in order to achieve this algorithm known as Code texted Levenshtein algorithm is employed. Code texted Levenshtein algorithm calculates the least number of edit operations that are necessary to modify one string to obtain another string. The Code texted Levenshtein distance between two words is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other. This can be implemented as dynamic programming approach. A matrix (m, n) cell is used and the Code texted Levenshtein distance between m-character prefix of one with n-prefix of other word. The weight or cost of each modification is calculated .Once the threshold value is met it ensures leakage of sensitive information and the owner is alerted.

D. Sensitive Data Detection

Once the DLD checker checks the outsourced content, if any data leak is identified and hits threshold, it is reported to the data owner since every data owner maintains common access condition for every file that enables DLD to detect unauthorized outsourcing. This results in detecting who is trying to access data in an unauthorized way, to which they are trying to transfer and alert the owner.

V. RESULT DISCUSSION

In an existing Works the realizations of data-leak detection require the plaintext sensitive data. However, this requirement is undesirable, as it may threaten the confidentiality of the sensitive information. If a detection system is compromised, then it may expose the plaintext sensitive data. Shingle with Rabin fingerprint was used previously for identifying similar spam messages in a collaborative setting as well as collaborative worm containment virus scan and fragment detection. Privacy requirement does not exist in above models, because if a detection system is compromised, then it may expose the plaintext sensitive data. Iterations for sensitive data go undetected.

Our proposed model overcomes these issues. It enables the data owner to securely delegate the content-inspection task to DLD providers without exposing the sensitive data. To prevent the DLD provider from gathering exact knowledge about the sensitive data, the collection of potential leaks is composed of real leaks and noises. It is the data owner, who post-

processes the potential leaks sent back by the DLD provider and determines whether there is any real data leak. In post process data owner can preserve sensitive data all time. data owner can take action against malicious outsourcing agents. Overall the system will give reliability to stack holders.

The screenshots of current privacy model is shown. all background data are transferred using Base 64 algorithm. Intentional data leaks are detected using Code Texted Levenshtein algorithm and alerts are made to the owner.

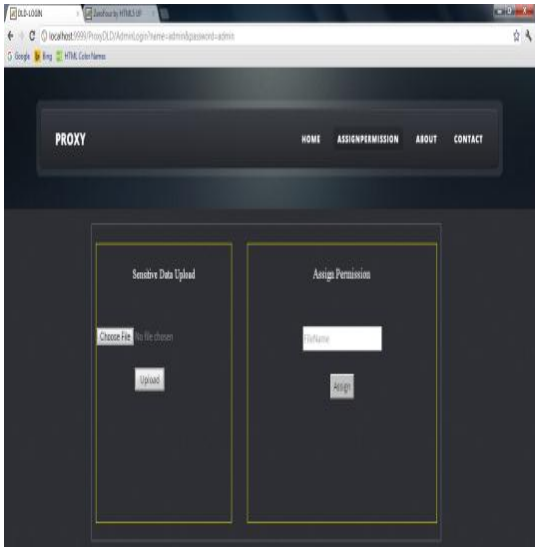

Figure 4: Generating Sensitive Data.

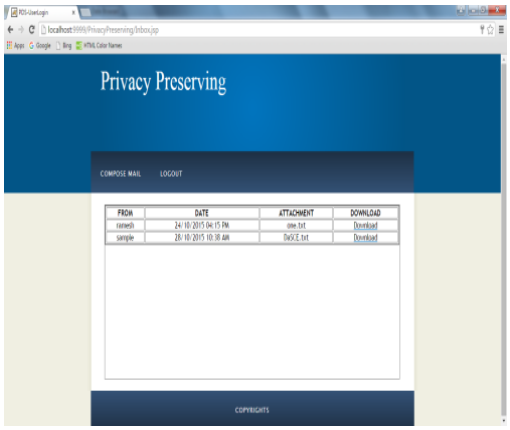

Figure 5: Privacy Preserving.

VI. CONCLUSION AND FUTURE WORK

Preventing sensitive data from being compromised is an important and practical research problem. The privacy goal is quantified and restricting the probability that the DLD provider identifies the exact value of the sensitive data.

In this work, a system that uncovers data leaks of personal information in transmissions, without the knowledge of data owner, by unauthorized persons is developed. The system discovers leaks even when the structure of information being leaked was previously unknown modified form, and does so without requiring a deep or intrusive instrumentation of the computing system and in a cost effective manner. Future work concentrates on security issues to be faced for the healthcare Internet of Things [IoT] realization in the real world.

REFERENCES

[1] S. Ananthi, M. Sadish Sendil, and S. Karthik, "Privacy preserving keyword search over encrypted cloud data," in Advances in Computing and Communications (Communications in Computer and Information Science), vol. 190. Berlin, Germany: Springer-Verlag, 2011,pp. 480–487.

[2] K. Borders and A. Prakash.(2009), "Quantifying information leaks in outbound web traffic," in Proc. 30th IEEE Symp. Secure. Privacy, pp. 129–140

[3] B. M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos, "SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics," in Proc. 19th USENIX Conf. Secur. Symp., 2010, p. 15.

[4] Carbunar and R. Sion.(2010), "Joining privately on outsourced data," in Secure Data Management (Lecture Notes in Computer Science), vol. 6358. Berlin, Germany: Springer-Verlag, , pp. 70–86.

[5] R. Chen, B. C. M. Fung, N. Mohammed, B. C. Desai, and K. Wang.(2013),"Privacy-preserving trajectory data publishing by local suppression," Inf.Sci., vol. 231, pp. 83–97.

[6] J. Croft and M. Caesar.(2011), "Towards practical avoidance of information leakage in enterprise networks," in Proc.

[7] *T. W. Fawcett,* "ExFILD: A tool for the detection of data exfiltration using entropy and encryption characteristics of network traffic," M.S. thesis, Dept. Elect. Comput. Eng., Univ. Delaware, Newark,DE, USA, 2010.

[8] *G. Jagannathan and R. N. Wright,* "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," in Proc. 11th ACMSIGKDD Int. Conf. Knowl. Discovery Data Mining, 2005, pp. 593–599.

[9] Y. Jang, S. P. Chung, B. D. Payne, and W. Lee.(2010), "Gyrus: A framework for ser-intent monitoring of text-based networked applications," in Proc.

[10] Jiang, X. Wang, and D. Xu. (2010),"Stealthy malware detection and monitoring through VMM-based ACM Trans. Inf. Syst. Secure., vol. 13, no. 2, , p 12.

[11] Jos´eMar´ıaG´omez-Hidalgoy,Jos´e Miguel Mart´ın-Abreuy,Javier Nievesx, Igor Santosx, Felix Brezox, and Pablo G. Bringasx "Data Leak Prevention through Named Entity Recognition",2010,

[12] Jung, A. Sheth, B. Greenstein, D. Wetherall, G. Maganis, and T. Kohno.(2008), "Privacy oracle: A system for finding application leaks with black box differential testing," in Proc. 15th ACM Conf. Computer Communication. Security., pp. 279–288.Kapravelos,

[13] J. Kleinberg, C. H. Papadimitriou, and P. Raghavan.(2001), "On the value of private information," in Proc. 8th Conf. Theoretical Aspects Rationality Knowl., pp. 249–257.

[14] P-C.Lin, Y.-D. Lin, Y.-C. Lai, and T.-H. Lee.(2008), "Using string matching for deep packet inspection," IEEE Comput., vol. 41, no. 4, pp. 23–28.

[15] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in Proc. 29th IEEE Conf. Comput. Commun., Mar. 2010, pp. 1–5.

[16] C. P. Mayer, "Bloom filters and overlays for routing in pocket switched networks," in Proc. 5th Int. Student Workshop Emerg. Netw. Experim.Technol., 2009, pp. 43–44.

[17] A. Nadkarni and W. Enck, "Preventing accidental data disclosure in modern operating systems," in Proc. 20th ACM Conf. Comput. Commun.Secur., 2013, pp. 1029–1042.

[18] Olive Qing Zhang, Ryan K L Ko, Markus Kirchberg, Chun HuiSuen, Peter Jagadpramana, Bu Sung Lee "How to Track Your Data:Rule-Based Data Provenance Tracing Algorithms",Hewlett-PackardLaboratories, Singapore 2012.

[19] Y. Shoshitaishvili, M. Cova, C.Kruegel, and G. Vigna,Revolver.(2013): An automated approach to the detection of evasive web-based malware," in Proc. 22nd USENIX Secure Symp., pp. 637–652.

[20] R. Paulet, and E. Bertino.(2013), Private Information Retrieval (Synthesis Lectures on Information Security, Privacy, & Trust). San Rafael, CA, USA: Morgan & Claypool Pub.,

[21] X. Shu and D. Yao.(2012), "Data leak detection as a service," in Proc. 8th Int.Conf. Secur. Privacy Commun. Netw., pp. 222–240.

[22] B. Wang, S. Yu, W. Lou, and Y. T. Hou.(2014), "Privacy-preserving multi-key word fuzzy search over encrypted data in the cloud," in Proc. 33th IEEE Conf. Comput. Commun., pp. 2112–2120.

[23] A. C.-C. Yao, "How to generate and exchange secrets," in Proc. 27th Annu. Symp. Found. Comput. Sci., 1986, pp. 162–167.

[24] D. Yao, K. B. Frikken, M. J. Atallah, and R. Tamassia, "Private information: To reveal or not to reveal," ACM Trans. Inf. Syst. Secur.,vol. 12, no. 1, 2008,

[25] H. Yin, D. Song, M. Egele, C. Kruegel, and E. Kirda, "Panorama:Capturing system-wide information flow for malware etection andanalysis," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007,pp. 116–127.

[26] X. Yi, R. Paulet, and E. Bertino, Private Information Retrieval (Synthesis Lectures on Information Security, Privacy, & Trust). San Rafael,CA, USA: Morgan & Claypool Pub., 2013.

[27] X. Yi, M. G. Kaosar, R. Paulet, and E. Bertino, "Single-database private information retrieval from fully homomorphic encryption," IEEE Trans.Knowl. Data Eng., vol. 25, no. 5, pp. 1125–1134, May 2013.

[28] Xiaokui Shu, Danfeng Yao," Privacy-Preserving Detection of Sensitive Data Exposure", Ieee Transactions on Information Forensics And Security, Vol. 10, No. 5, MAY 2015.