e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

DATA LEAKAGE DETECTION USING FLASK

Himanshu Taiwade*[1], Uday Singh*[2]

*[1]Project Guide, Department of Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur, Maharashtra, India.

*[2]Student, Department of Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur, Maharashtra, India.

ABSTRACT

We propose data allocation strategies across the agents that improve the probability of identifying leakages. These methods do not rely on alterations of the released data e.g., watermarks. In some cases, we can also inject "realistic but fake" data records to further improve our chances of detecting leakage and identifying the guilty party. The idea of modifying the data itself to detect the leakage is not a new approach. Generally, the sensitive data are leaked by the agents, and the specific agent responsible for the leaked data should always be detected at an early stage. Thus, the detection of data from the distributor to agents is mandatory. This project presents a data leakage detection system using various allocation strategies that assess the likelihood that the leaked data came from one or more agents For secure transactions, allowing only authorized users to access sensitive data through access control policies shall prevent data leakage by sharing information only with trusted parties and also the data should be detected from leaking through adding the fake record`s in the data set and which improves the probability of identifying leakages in the system. Then, finally, it is decided to implement this mechanism on the cloud server.

Keywords: Cloud Computing, Data Leakage, Data Security, SHA, Third Party Agent, Fake Object

I. INTRODUCTION

In today's era, most of the leaked sensitive data records have increased dramatically. Data leakage designates the unauthorized transmission of sensitive data or information from within an organization to an external destination where the confidentiality of information is compromised. A prevalent approach is to monitor the data in storage and transmission to expose sensitive information. Withal it considers all data as sensitive and performs detection operations for all those data. However, this makes the detection process arduous, and detection time increases. In addition, the data owner may be required to provide a detection report to the DLD provider. However, there is the possibility that the provider can read the sensitive data. To minimize the leakage of sensitive data, the organization needs to avert clear text-sensitive data from appearing in the storage. A screening implement is used to scan the files. Consequently one needs an incipient data detection solution that sanctions providers to scan the content for leaks without learning information. Ergo one needs methods that give precise detection with a minutely diminutive number of erroneous alarms under sundry leak scenarios.

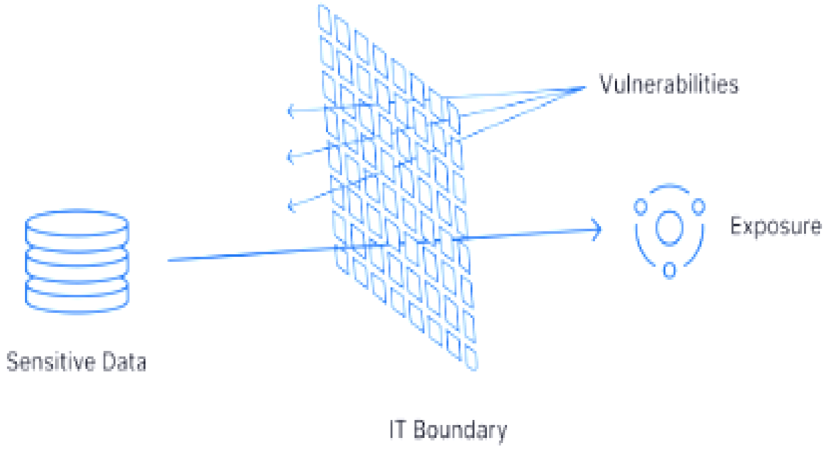


Figure 1: Data Leak Mechanism

Human mistakes play a paramount role in the cause of data loss among sundry data leaks. There are sundry methods to detect data leaks caused by human mistakes and avert the data by engendering a vigilante. Among sundry approaches, monitoring the data that is transmitted for exposure to sensitive information is mundane. Withal, it considers all data as sensitive and performs detection operations for all those data. However, this makes the detection process arduous, and the detection time increases. So there is a desideratum for incipient data detection solutions that sanction providers to scan the content for leaks without learning information. Ergo one needs methods that give precise detection with a minutely diminutive number of erroneous alarms under sundry leak scenarios and the result shows that the method ameliorates the detection time.

II. METHODOLOGY

P. Papadimitriou et al., [1] we study the following quandary: A data distributor has given sensitive data to a set of suppositious trusted agents (third parties). Some of the data is leaked and found in an unauthorized place (e.g., on the web or somebody's laptop). The distributor must assess the likelihood that the leaked data emanated from one or more agents, as opposed to having been independently amassed by other betokens. We propose data allocation strategies (across the agents) that ameliorate the probability of identifying leakages. These methods do not rely on alterations of the relinquished data (e.g., watermarks). In some cases, we can withal inject “realistic but fake” data records to further amend our chances of detecting leakage and identifying the culpable party.

V. Srivastava et al., [2] data Leakage is the most astronomically immense issue, in this digital world. Immensely colossal organizations keep consolidated data of personal information of employees, users, customers, clients, etc. Even though there are many algorithms utilized for privacy preservation to defend sensitive or confidential data, data leakage becomes an uncontrolled threat in the digital world. This research purports to analyze the performance of data leakage detection systems utilized in the information retrieval system in web applications. In this paper, authors consider the models such vector space model and the Interaction information retrieval model. The conception is to check the semantically kindred documents on the Web for leaked data. The result inferred that the Interaction Information Retrieval Model is better than the Vector Space Model.

N. Kumar et al.,[3] in recent years internet technologies have become the backbone of any business organization. These organizations use this facility to improve their efficiency by transferring data from one location to another. However, there are several threats in transferring critical organizational data as any culprit employee may publicize this data. This problem is known as the data leakage problem. In the proposed work, we are suggesting a model for the data leakage problem. In this model, the authors aim to identify the culprit who has leaked the critical organizational data.

S. Natesan et al.,[4] this paper propose a novel solution for the detection of data leaks utilizing dynamic and static analysis. A dataset of Android applications mapped to their data leaks, sanctions, activities, library classes, and methods is built into this approach. During dynamic analysis, the solution uses runtime sanctions and logcat information to detect the data leak type and calculate their probability of occurrence utilizing a k-most proximate neighbor's kindred algorithm. This approach further categorizes the leaks on the substructure of the peril level associated with that application.

D. Jyothirmai et al.,[5] this paper proposes strategies for allocating data across agents to ameliorate the probability of identifying leaks in the event of data leakage. The proposed methods do not rely on altering the relinquished data like watermarks. Instead, they fixate on distributing the data in a way that makes it more facile to identify the source of the leak. The paper highlights that data leakage is a paramount quandary that can lead to a loss of mazuma, damage to reputation, and more. It can occur due to security susceptibilities, poor data bulwark practices, or fortuitously by a utilizer. Consequently, it's essential to take measures to eschew data leakage and bulwark data. Utilizing the proposed strategies, the distributor can ameliorate the chances of identifying the source of the leak and taking felicitous action to avert data leakage.

S. K. Nayak et al.,[6] in this paper, firstly authors identify the challenges and impact regarding data leakage for a user in a smart home environment. Secondly, the authors proposed a secure framework for data communication and storage. Thirdly, the proposed model implementation and security analysis are given in detail. Lastly, a security comparison statement is given with some existing related work which shows the proposed model is more secure in terms of data leakage.

T. Rocha et al.,[7] in this work we are addressing this quandary by engendering a modified version of Tizen, called TTizen, that modifies the Tizen Web Runtime to integrate dynamic taint tracking, with which we can track sensitive information that is being leaked, even if the information is obfuscated, and admonish the users. From our erudition, this is the first archetype that integrates this kind of technique into Tizen and tracks web applications in mobile contrivances. The results show that TTizen is a promising approach that can be ameliorated and used to detect data leakage in IoT contrivances.

F. Zou et al., [8] DNS data leakage exhibits low traffic volume and periodicity, which is thoroughly different from DNS tunnels with bi-directional data exchange and high traffic volume. In this paper, a detection model designated as LSTM-AE is proposed. LSTM-AE integrates LSTM-predicated time-series characterization and unsupervised auto encoder to detect data leakage malware through DNS traffic. Experimental results show the detection performance of LSTM-AE is better than other ML-predicated methods and several unknown maleficent domains cognate data leakage have been detected with genuine-world DNS traffic.

III. MODELING AND ANALYSIS

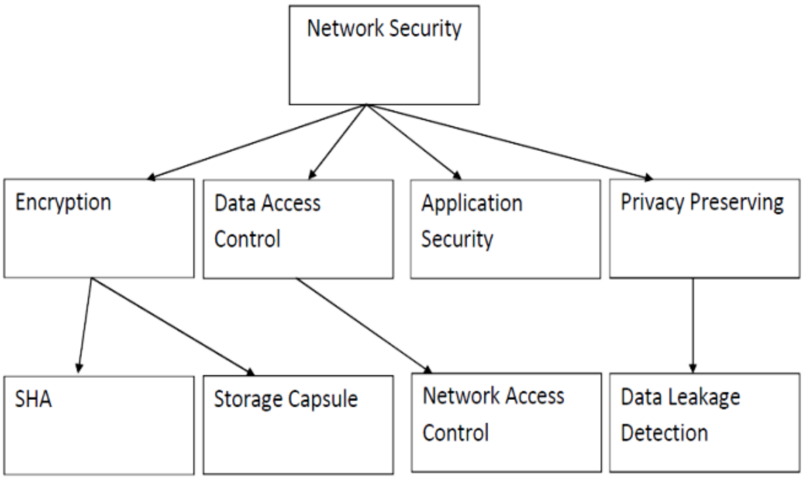


Figure 2: Data Flow Diagram

In the proposed system, host host-assisted mechanism is used which checks the frequency of occurrence of data. Highly differentiated values are considered sensitive and fingerprints are generated for them. Repeated values are ignored in this method. The statistical approach is used to generate sensitive data and it is stored in the table. The fingerprints are generated by data leak detection (DLD) providers and identify potential leaks by matching the fingerprints. The potential leak consists of real leaks and noises so that no one can get exact information about the sensitive data. The data owner post-processed data is sent by the DLD provider to check where there is any leak in the sensitive data. The objectives are to improve the detection time and to improve the detection of sensitive packets.

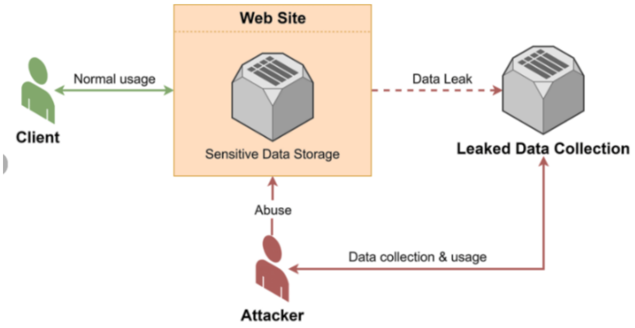


Figure 3: Data Leakage Flow Diagram

Building a data leak detection model using Flask and Machine Learning (ML) involves several steps, from data preprocessing and model training to integrating the model into a Flask web application for real-time monitoring. Below are the steps to create the flow of the project:

1.1 Define Data Leak Features:

- Identify features relevant to data leak detection. Examples include source and destination IP addresses, timestamps, data volume, and patterns of access.

1.2 Data Collection and Preprocessing:

- Collect labeled data for training the model. This should include instances of normal data flows and instances of data leaks.
- Preprocess the data by cleaning, handling missing values, and converting it into a format suitable for ML algorithms.

3.3 Feature Engineering:

- Extract meaningful features from the data to improve the model's performance.
- Transform categorical features into numerical representations if needed.

3.4 Split Data into Training and Testing Sets:

- Split the labeled dataset into training and testing sets to evaluate the model's performance.

3.5 Choose ML Algorithm:

- Select an ML algorithm suitable for anomaly detection. Common choices include Isolation Forests, One-Class SVM, or Autoencoders.

3.6 Train the Model:

- Train the selected ML model using the labeled training dataset.

3.7 Real-time Monitoring:

- Implement a mechanism to monitor incoming data in real time and use the trained model for anomaly detection.

3.8 Alert Mechanism:

- Develop an alert system within Flask to notify administrators when potential data leaks are detected.

3.9 User Authentication and Authorization:

- Implement user authentication and authorization mechanisms within the Flask application to control access.

3.10 Logging and Auditing:

- Set up logging to record activities and potential data leak events.
- Regularly audit the logs for suspicious activities.

3.11 Testing and Validation:

- Test the Flask application and ML model thoroughly, validating their accuracy and effectiveness.

IV. RESULTS AND DISCUSSION

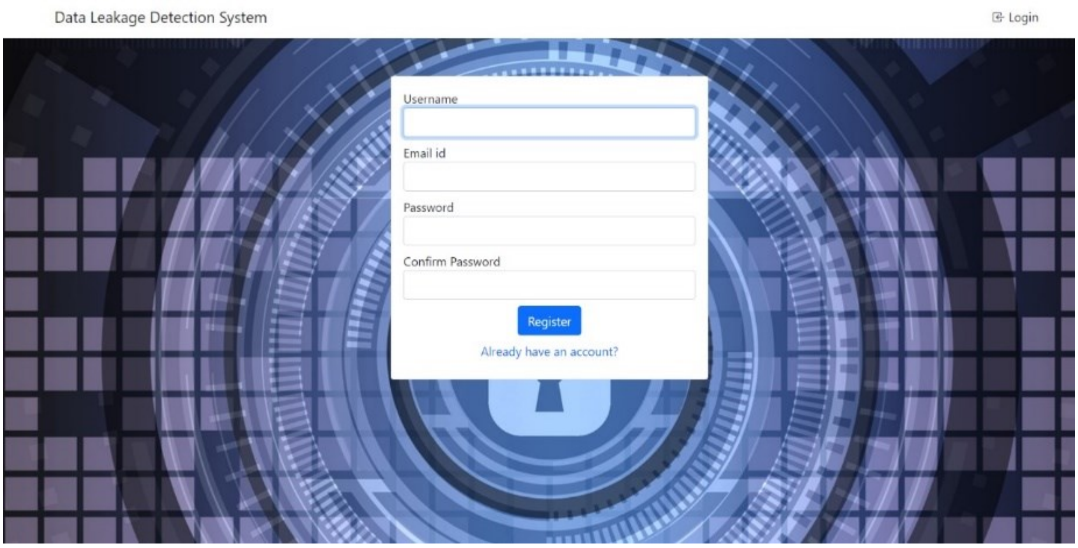
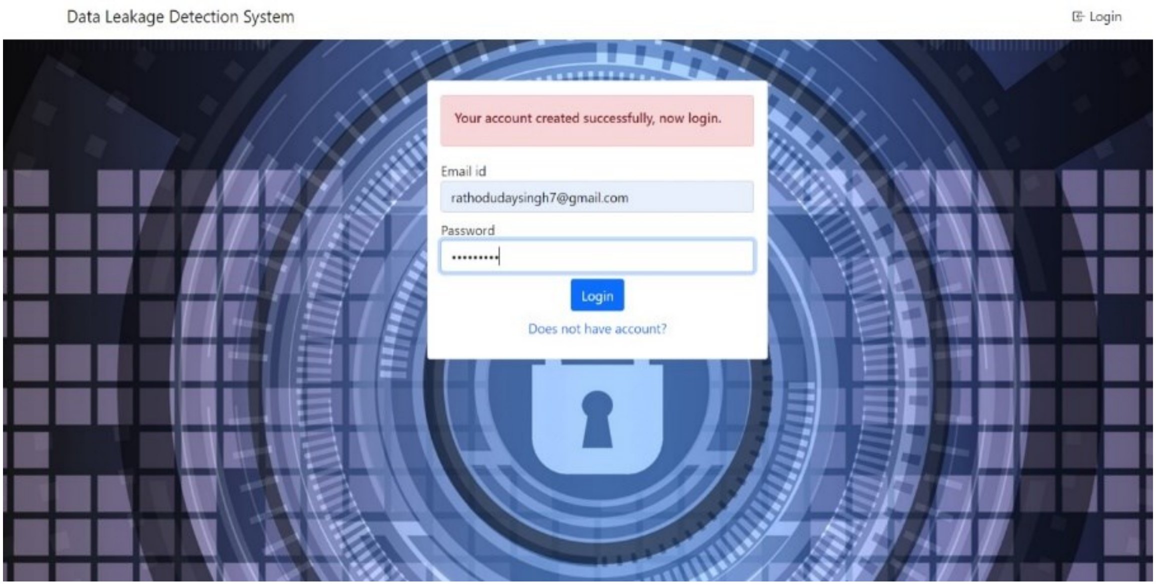


Figure 4.1 User Registration Panel



Your account created successfully, now login.

Email id: rathodudaysingh7@gmail.com

Password: •••••••••

Figure 4.2 User Login Panel

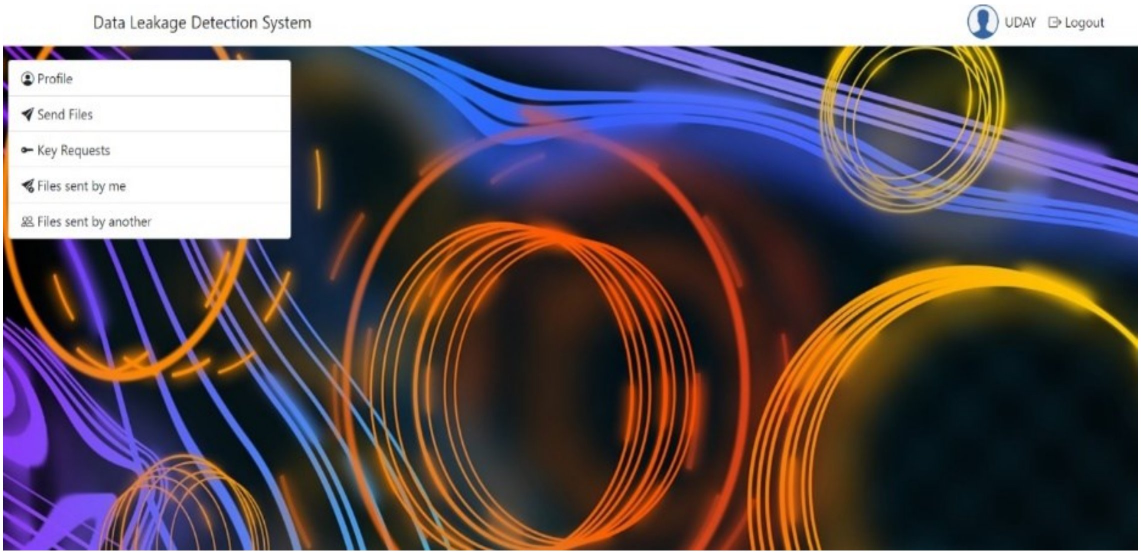
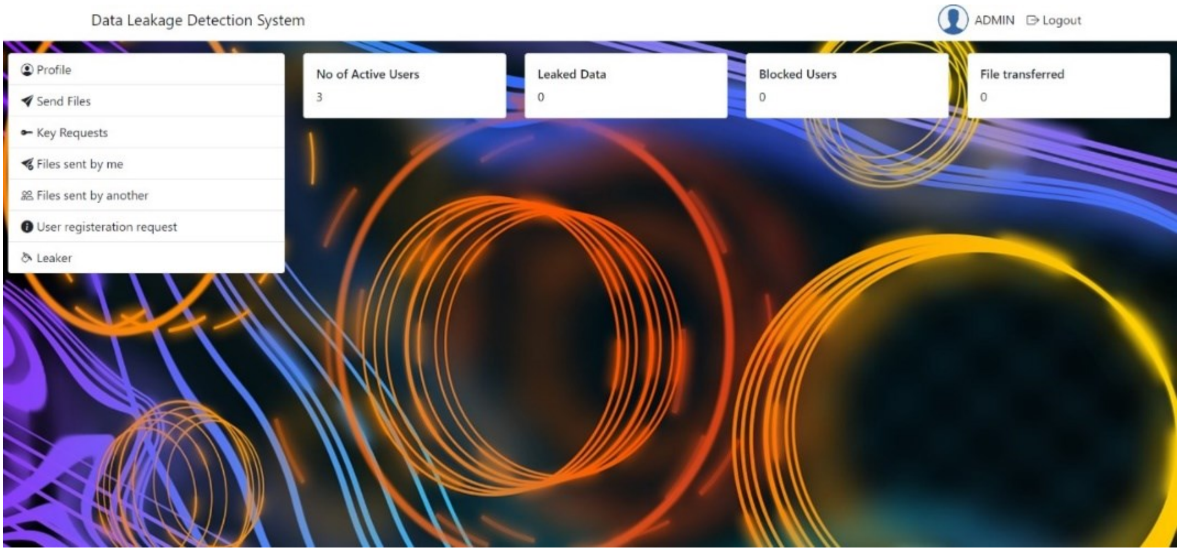


Figure 4.3 User Dashboard



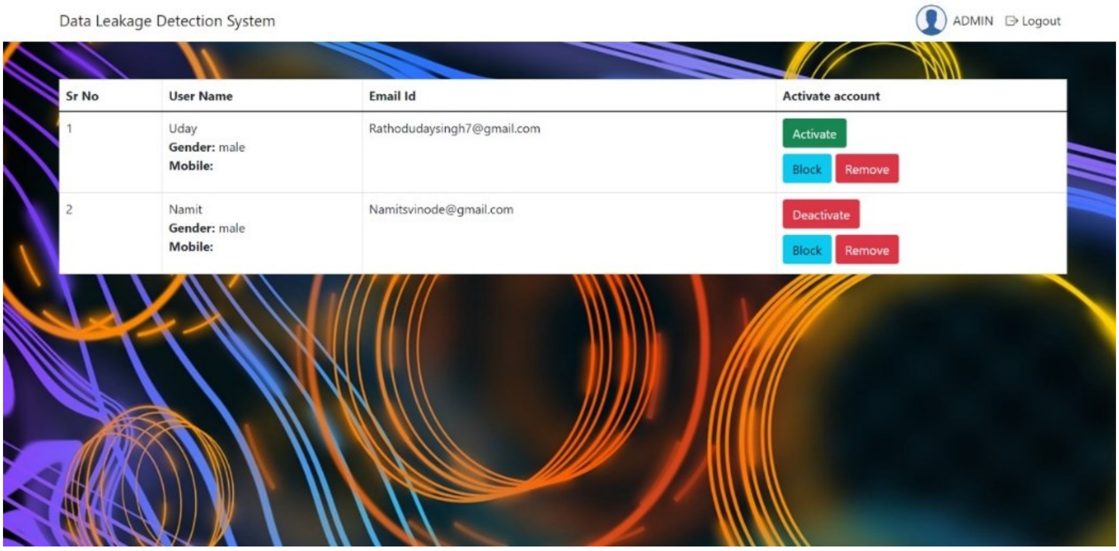
No of Active Users
3

Leaked Data
0

Blocked Users
0

File transferred
0

Figure 4.4 Admin Dashboard



Sr No	User Name	Email Id	Activate account
1	Uday **Gender:** male **Mobile:**	Rathodudaysingh7@gmail.com	Activate Block Remove
2	Namit **Gender:** male **Mobile:**	Namitsvinode@gmail.com	Deactivate Block Remove

Figure 4.5 Active Users Details

V. CONCLUSION

Data leakage is a silent type of threat. Your employee as an insider can intentionally or accidentally leak sensitive information. This sensitive information can be electronically distributed via e-mail, Web sites, FTP, instant messaging, spreadsheets, databases, and any other electronic means available – all without your knowledge. To assess the risk of distributing data two things are important, where first one is a data allocation strategy that helps to distribute the tuples among customers with minimum overlap and the second one is calculating guilt probability which is based on the overlapping of his data set with the leaked data set.

VI. REFERENCES

[1] P. Papadimitriou and H. Garcia-Molina, "Data Leakage Detection," in IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 1, pp. 51-63, Jan. 2011, doi: 10.1109/TKDE.2010.100.

[2] V. Srivastava, A. Majumdar and J. A, "Performance Analysis of Data Leakage Detection System," 2022 3rd International Conference for Emerging Technology (INCET), Belgaum, India, 2022, pp. 1-5, doi: 10.1109/INCET54531.2022.9824111.

[3] N. Kumar, V. Katta, H. Mishra and H. Garg, "Detection of Data Leakage in Cloud Computing Environment," 2014 International Conference on Computational Intelligence and Communication Networks, Bhopal, India, 2014, pp. 803-807, doi: 10.1109/CICN.2014.172.

[4] S. Natesan, M. R. Gupta, L. N. Iyer and D. Sharma, "Detection of Data Leaks from Android Applications," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 326-332, doi: 10.1109/ICIRCA48905.2020.9183066.

[5] D. Jyothirmai, B. Vineela, R. Pitchai, B. Dinesh, M. Indhu and A. G. Avinash, "Data Leakage Detection Using Secret Key Exchange," 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 2023, pp. 218-222, doi: 10.1109/ICECAA58104.2023.10212102.

[6] S. K. Nayak, S. Keshari Swain, B. K. Mohanta and B. Kumar Paikaray, "Secure Framework for Data Leakage Detection and Prevention in IoT Application," 2022 IEEE 2nd International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), Gunupur, Odisha, India, 2022, pp. 1-6, doi: 10.1109/iSSSC56467.2022.10051336.

[7] T. Rocha et al., "Data leakage detection in Tizen Web applications," 2016 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 2016, pp. 608-614, doi: 10.1109/PST.2016.7906994.

[8] F. Zou, Y. Ren, J. Zhu and J. Tang, "Detecting Data Leakage in DNS Traffic based on Time Series Anomaly Detection," 2021 IEEE 23rd Int Conf on High-Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big

Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Haikou, Hainan, China, 2021, pp. 503-510, doi: 10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00090.

[9] X. Wang, Y. Zhang, Z. Li, "Anomaly-based Data Leak Detection in IoT Systems", IEEE Internet of Things Journal, 2020

[10] M. El-Halawany, A. Ali, M. Youssef, "A Survey of Data Leak Detection and Prevention Techniques", International Journal of Computer Applications, 2018

[11] M. U. Akram, M. U. Ilyas, A. Ahmed, "Towards Automated Data Leak Detection in Enterprise Networks", 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), 2019

[12] B. M. Al-Shammari, S. Dey, "Enhanced Data Leak Detection in Cloud Computing using Blockchain Technology", 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2018

[13] S. W. Kim, S. Y. Lee, J. S. Lee, "Data Leak Detection in Large-Scale Networks using Convolutional Neural Networks", 2018 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2018

[14] A. K. Sharma, M. K. Sharm, "Data Leak Detection and Prevention in Cloud Computing using Machine Learning Techniques", 2018 2nd International Conference on Inventive Systems and Control (ICISC), 2018

[15] M. A. Qadir, A. S. M. G. Subramaniyaswamy, "A Survey on Data Leak Detection and Prevention Techniques", International Journal of Computer Science and Information Technologies, 2016