AIJFR

Advanced International Journal for Research

E-ISSN: XXXX-XXXX editor@aijfr.com

Volume 2, Issue 3, May – June 2024

Detection and Safeguarding of Information Leakage Utilizing RSA Encryption in Cloud Computing

Saary Mega Utaamy

Research Scholar, UIN Mahmud Yunus Batusangkar, Indonesia

Abstract

Cloud computing has transformed the way businesses handle data storage and exchange. A hierarchical structure involving multiple users with varying access levels continuously shares and uploads large datasets to the cloud. As organizations increasingly adopt cloud-based systems for their data storage needs, ensuring a secure and efficient data access framework has become a critical focus of research.

Current approaches in this field often involve packet capture during the VM migration process to identify potential data leakage incidents. However, these methods typically rely on unencrypted data and may not fully capture the complexities encountered in real-world scenarios. In contrast, the proposed system introduces a robust data security framework centered around RSA encryption. By leveraging asymmetric cryptography, RSA encryption significantly enhances data protection.

To further strengthen data transmission security, the system integrates RSA encryption with secure communication channels. Moreover, RSA-supported authentication mechanisms are implemented to ensure controlled user access. These advancements aim to mitigate security risks associated with data storage and transmission in cloud environments, thereby enhancing overall system resilience and integrity.

Keywords: Content inspection involves examining content for various purposes such as data leakage detection, employing dynamic programming and sampling algorithms.

1. Introduction

1.1content Inspection

In today's digital era, safeguarding sensitive data is paramount. Content inspection, a critical component of data security, plays a pivotal role in identifying, analyzing, and managing content transmitted through diverse communication channels. As organizations increasingly rely on digital platforms for communication, collaboration, and data sharing, content inspection techniques have become indispensable in mitigating risks like unauthorized access, data leakage, and malicious activities. This process involves systematically scrutinizing digital content—such as text, images, files, and other data forms—to ensure adherence to security policies, regulatory requirements, and ethical standards. This proactive approach empowers organizations to detect potential threats, enforce data protection measures, and maintain the integrity of their digital assets.

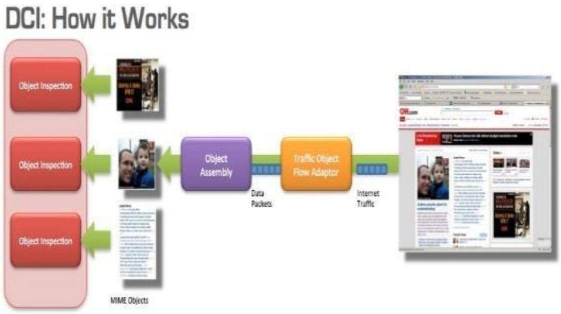

Figure 1 Content Inspection

1.2 Data Leakage Detection

In an era marked by exponential growth in digital data and interconnected systems, protecting sensitive information is of utmost importance. The rise of cyber threats and the increasing sophistication of malicious actors underscore the critical need for robust measures to detect and prevent data leakage. Data leakage, which entails the unauthorized exposure of confidential information, poses significant risks to individuals, organizations, and society at large. This survey explores the field of data leakage detection, examining various techniques and strategies employed to identify, mitigate, and respond to potential data breaches. As organizations store and exchange vast amounts of sensitive data across networks and platforms, the threat of unauthorized access and disclosure remains pronounced. Detecting data leakage involves actively monitoring and analyzing data traffic to detect anomalies, suspicious patterns, or unauthorized attempts to access data.

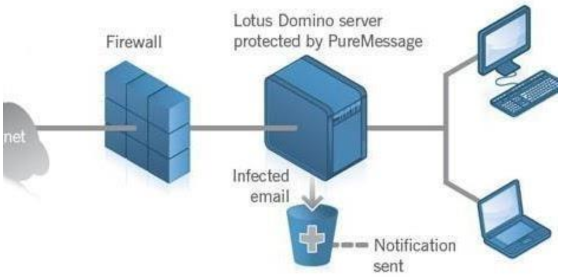

Figure 2 Data Leakage Detection

1.3 Dynamic Programming

Dynamic programming is a powerful optimization technique widely used in computer science and mathematics. It is instrumental in addressing problems characterized by overlapping subproblems and optimal substructure. This method originated as a way to efficiently tackle complex problems by breaking them down into smaller, more manageable subproblems. Its applications span various fields including algorithm design, operations research, and artificial intelligence. At the heart of dynamic programming is its ability to store and reuse solutions to subproblems, thereby avoiding redundant computations and significantly improving algorithm efficiency. This survey explores the foundational principles, methodologies, and practical applications of dynamic programming, offering insights into its broad relevance and profound impact.

1.4 Sampling Algorithm

In computer science and data analysis, sampling algorithms are pivotal for extracting valuable insights from vast datasets. As data volumes continue to grow exponentially, the efficiency of processing and analyzing such data becomes increasingly critical. Sampling algorithms address this challenge by providing methods to select a representative subset of data points for analysis. This approach significantly reduces computational complexity while ensuring statistical validity. Sampling algorithms find applications across diverse domains including data analytics, machine learning, quality assurance, and market research. The fundamental idea is to select a subset, or sample, from a larger dataset that accurately represents the dataset's characteristics. This enables analysts and researchers to draw conclusions, make predictions, and gain insights into the broader population without needing to analyze every individual data point.

2. Literature Survey

A literature review is an essential component providing a comprehensive analysis of previously published works, conference proceedings, and research papers focusing on data leakage detection and protection. The reviewed literature is categorized based on primary themes, techniques employed, datasets used, and experimental outcomes.

Isabel Herrera Montano et al. (2022) introduced enhanced data leakage detection and protection systems integrating advanced technologies such as encryption, VFS, hash functions, and digital rights management policies. Encryption and machine learning emerged as the most commonly utilized techniques in the examined Data Loss Prevention (DLP) tools.
Abdel-Rahman Al-Ghuwairi et al. (2023) proposed a collaborative feature selection model that integrates time series analysis techniques with anomaly detection and causality tests to address misleading connections between anomalies and attacks.

Unnati Komal et al. (2023) devised mechanisms to prevent unauthorized data access and ensure data privacy by detecting anomalies in user behavior using machine learning algorithms. They also classified data based on sensitivity levels and applied access control policies accordingly.
Mohankumar et al. (2023) introduced a strategy involving the addition of realistic but false records to enhance data distribution strategies and improve data security through encryption and perturbation techniques.

Adinarayana et al. (2023) proposed a novel cloud storage scheme employing a trustworthy audit server for data preprocessing and upload, utilizing anxiety techniques to modify data sensitivity before transmission.
Vijaya Balpande et al. (2023) discussed data distribution strategies aimed at increasing the likelihood of detecting leaks without altering publicly available data, occasionally incorporating realistic but fictitious data entries.
Rishabh Singh et al. (2022) presented a model involving packet capture during data transmission via Third Party Applications (TPA), ensuring data protection through secure key exchange.
Ishu Gupta et al. (2022) reviewed various data protection mechanisms and proposed innovative solutions while discussing research gaps and future directions in the field.

Prisca I. Okochi et al. (2021) implemented ASP.net MVC and Microsoft SQL Server Management Studio for backend data protection, incorporating an Audit trail/Transaction log mechanism to monitor computing environment activities.
Ehab Zaghloul et al. (2020) introduced the Privilege-based Multilevel Organizational Data-sharing scheme (P-MOD), combining privilege-based access structures with attribute-based encryption mechanisms to securely manage and share large datasets, assuming security against adaptively chosen plaintext attacks.

3. Existing System

Data constitutes a critical asset for organizations, encompassing valuable intellectual property such as customer records, financial data, patient information, personal credit card details, and other pertinent information relevant to their operations, management, or industry. Safeguarding this sensitive data is paramount as any leakage can incur substantial costs and pose significant challenges. Information leakage refers to the unauthorized exposure of individual or organizational data to third parties, resulting in direct financial losses and operational disruptions. Moreover, leaked information can introduce vulnerabilities and unauthorized modifications.

To mitigate these risks, organizations must implement robust measures to prevent unauthorized data leaks. Various methods have been developed over time to address this issue, as explored in this survey. This paper investigates recent techniques and proposes an innovative approach based on Sampling algorithm.

4. Proposed System

The proposed system integrates a comprehensive Data Owner and User Module that facilitates secure access through user login. Users are required to input their user type, username, and password, ensuring robust system security for both data owners and regular users. New users can register by submitting necessary details such as user type, username, password, and email ID. This registration process ensures that access is granted only to authorized individuals. The Cloud Server Module acts as a centralized repository responsible for securely managing and storing critical information, including user profiles, encryption keys, and files. Additionally, the Secure Access Control - Data Owner Module empowers data owners to generate robust RSA encryption keys to protect stored data effectively. This capability allows data owners to securely store and manipulate information on the cloud server, ensuring restricted access to authorized personnel.

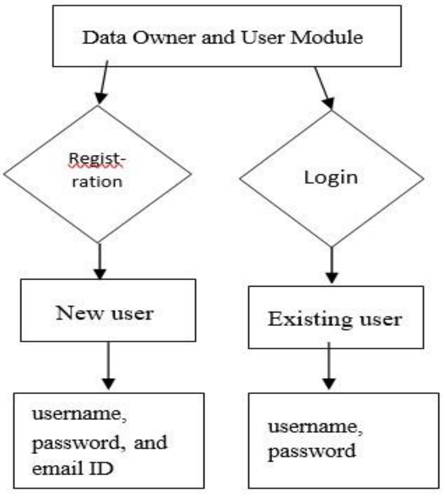

Figure 3. Module diagram

5. Module Descriptions

Data Owner and User Module:

Login: Users, whether they are data owners or regular users, access the system securely by providing their user type, username, and password. This login process ensures a secure entry into the system.

Registration: New users can register by providing essential information, including user type, username, password, and email ID. This module guarantees that only authorized individuals gain access to the system.

5.2 Cloud Server Module:

The Cloud Server Module manages and securely stores critical information on the cloud server, including user details, encryption keys, and stored files. Serving as a central repository for data storage, it ensures efficient and secure information management.

A cloud server, part of the IaaS model, is a virtualized compute server accessed remotely, offering functionalities similar to physical servers. It plays a pivotal role in cloud computing models like IaaS, PaaS, and SaaS, created by virtualizing physical servers.

5.3 Secure Access Control - Data Owner Module:

The Secure Access Control - Data Owner Module enables data owners to generate encryption keys, ensuring robust RSA encryption for securing stored data. This enhances data protection significantly. Additionally, data owners can securely store their data on the cloud server, restricting access to authorized individuals.

5.4 Secure Access Control - User Module:

The Secure Access Control - User Module allows regular users to access information about files stored on the cloud server, including file names, types, and relevant metadata. Users can request access to specific files, which undergo authentication and verification processes before access is granted. This approach ensures data security throughout the file access process, enabling users to securely retrieve and view authorized file content.

6. System Implementation

The implementation phase within the proposed framework involves deploying and executing the developed software in a practical, real-world setting. This stage includes installing system components, configuring databases, and integrating with existing infrastructure. It encompasses transferring data, system settings, and user accounts from the testing environment to the live production environment. During implementation, thorough training programs are conducted to acquaint end-users with the system's functionalities and ensure a smooth transition. System administrators oversee the deployment process, addressing any unforeseen challenges or technical issues promptly. Furthermore, ongoing support mechanisms are established to offer assistance after implementation, ensuring a seamless and successful integration of the proposed system into the operational environment.

7. Algorithm Implementation

The encryption key is maintained confidentially and utilized for decryption. This system guarantees secure communication across insecure channels by enabling encryption using the public key and decryption using the private key. key is kept confidential and used for decryption. This system ensures secure communication over insecure channels by enabling encryption with the public key and decryption with the private key.

8. **Result Analysis**

algorithm	RSA	Existing algorithm
Precision	0.73	0.8
Recall	0.7	0.81
F score	0.74	0.84
Accuracy	0.71	0.8

Table 1. Comparison table

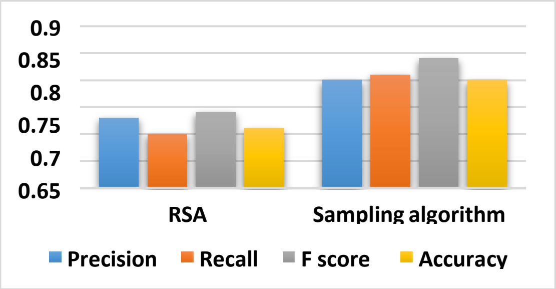

Figure 5. Comparison graph

When comparing RSA and a Sampling Algorithm, their performance metrics reveal similar results in terms of precision, recall, F score, and accuracy. The RSA algorithm shows strong performance with a precision of 0.73, recall of 0.7, F score of 0.74, and accuracy of 0.71. In contrast, the Sampling Algorithm outperforms RSA with higher metrics: accuracy of 0.8, recall of 0.81, F score of 0.84, and precision of 0.8.

The Sampling Algorithm demonstrates superior precision, recall, and F score values, indicating its capability to accurately identify positive cases and thereby improving overall accuracy compared to RSA. While both methods perform well, the Sampling Algorithm excels particularly in accurately classifying instances, making it suitable for applications where precision and recall are crucial for classification tasks.

9. Conclusion

In summary, the proposed system leverages RSA encryption within a client-server framework to provide comprehensive data security. By integrating secure key management, encrypted storage, and RSA-based authentication, the system establishes robust defenses against unauthorized access. Secure communication channels further enhance data transmission security, while auditing and monitoring features bolster accountability and threat detection. Regulatory adherence and regular system maintenance ensure a holistic approach to maintaining data integrity and confidentiality across diverse computing environments. Thoughtful input and output design enhances user interactions, ensuring a seamless experience. Through rigorous testing and implementation, the proposed framework not only meets specified requirements but also delivers a dependable and user-friendly software solution to tackle the evolving challenges of data protection and secure communication.

10. Future Work

To improve data security, the proposed system should explore advanced encryption technologies beyond RSA and include emerging cryptographic methods. Additionally, integrating machine learning algorithms for anomaly detection and adaptive security measures could greatly enhance threat detection capabilities. Continuous research and development efforts should focus on optimizing system performance, scalability, and responsiveness to ensure compatibility with evolving computing environments. Furthermore, investigating blockchain technology for decentralized and immutable data storage presents a promising avenue to enhance data integrity.

References

1. Herrera Montano et al. discuss techniques for data leakage protection and addressing insider threats using RSA encryption in a client-server model (2022).
2. Abdel-Rahman et al. explore intrusion detection in cloud computing through time series anomalies and machine learning (2023).

3. Komal et al. examine data leakage detection in cloud computing using machine learning methods (2023).
4. Mohankumar et al. propose a system for data leakage detection and security in cloud computing (2023).
5. Adinarayana et al. present strategies for data leakage detection in cloud computing environments (2023).
6. Balpande et al. review data leakage detection techniques utilizing cloud computing (2023).
7. Singh and Rajan discuss data leakage and security issues in cloud computing (2022).
8. Gupta and Singh present a comprehensive overview of data protection in communication and computing environments, emphasizing future directions (2022).
9. Chaudhary et al. propose optimized genetic algorithms and extended Diffie-Hellman methods for DOS-attack detection in cloud environments (2022).
10. Okochi et al. introduce an improved data leakage detection system for cloud computing environments (2021).
11. Alshammari and Alsubhi develop a reputation attack detector for trust evaluation in cloud services (2021).
12. Wang et al. focus on detecting and eliminating security risks in cloud computing projects (2021).
13. Sharma et al. investigate security risks and taxonomy in cloud computing environments (2021).
14. Al-Shehari and Alsowail utilize machine learning techniques for insider data leakage detection (2021).
15. Khushbu et al. propose a classification and distribution model for data leakage prevention and detection in cloud environments (2021).
16. Kiperberg et al. discuss an efficient hypervisor-based DLP system for cloud environments (2021).
17. Gupta and Singh introduce a guilty user identification model using distribution matrices for data leakage detection (2020).
18. Zaghloul et al. propose P-MOD for secure privilege-based data sharing in cloud computing (2020).
19. Gupta and Singh present an integrated approach for data leaker detection in cloud environments (2020).
20. Singh and Gupta develop an online information leaker identification scheme for secure data sharing (2020).