Data Leakage Detection in Cloud Computing

54 - Dhrithi Shetty
55 - Nidhi Shetty
56 - Sneha Shinde

Under the Guidance of
Prof. Monica Charate

Usha Mittal Institute of Technology
SNDT Women's University

Table of Contents

Introduction

- Cloud computing provides scalable data storage, but sensitive information faces risks of unauthorized access and data leaks.
- While encryption offers protection, it may not always be sufficient to trace the source of leaks or prevent misuse in shared environments.
- This project combines encryption along with watermarking and fake object insertion to enhance security by tracing leaks to specific users and detecting malicious activity.
- Real-time monitoring allows admins to investigate anomalies and respond to threats quickly, ensuring data protection and accountability in the cloud.

Problem Statement

To integrate watermarking and fake object insertion, we address the challenges of safeguarding sensitive data in cloud computing, which is increasingly vulnerable due to unauthorized access and potential leaks. While encryption provides protection, It lacks mechanisms to trace leaks back to specific users or detect malicious activities in shared environments. This project aims to overcome these limitations by detecting unauthorized access, tracing data breaches, and alerting administrators in real-time for timely intervention.

Literature Survey

No.	Paper Name	Author(s)	Remarks
1	Secure Data Storage and Sharing Techniques for Data Protection in Cloud Environments: A Systematic Review, Analysis, and Future Directions (2022)	Ishu Gupta Ashutosh Kumar Singh Chung-Nan Lee Rajkumar Buyya	This paper details a data leak detection approach using AES-128 encryption and unique watermarks to trace leaks. While effective in identifying guilty agents, it faces scalability and performance challenges.
2	Avoiding the data leakage and providing privacy to data in Networking (2016)	Madhavi Suryawanshi, Prof . Sarita Patil	The authors propose a system that enhances data leak detection security by introducing encryption and techniques to protect sensitive information during transmission and storage. AES is used to encrypt data. This algorithm is favoured for its speed and security.
3	A Survey on Data Breach Challenges in Cloud Computing Security: Issues and Threats(2017)	R.Barona, E.A.Mary Anita	The paper provides a broad overview of data breaches in cloud computing, discussing issues like data integrity and threat analysis. However, it is mostly theoretical and lacks practical solutions or empirical data

Literature Survey

4	Dynamic Data Leakage using Guilty Agent Detection over Cloud (2017)	Govinda. K, Divya Joseph	The authors discuss various data leakage detection methods, including watermarking techniques, and time-stamping. They propose a model that integrates these techniques to enhance data security in cloud computing.
5	An Analysis of Data Leakage and Prevention Techniques in Cloud Environments (2015)	T. Brindha, R. S. Shaji	The paper reviews data leakage issues and proposes preventive measures, including strategies. It discusses guilty agents, however, it is generic and not specific to cloud environments and lacks real-world implementations
6	Data Leakage Detection (2011)	Panagiotis Papadimitriou, Hector Garcia-Molina	This paper outlines a framework for detecting data leaks using strategic data allocation methods and additional techniques to enhance detection. It effectively balances leak detection with minimal disruption but faces challenges related to probabilistic accuracy and assumptions about user behavior.

Existing System

- Existing systems often rely on models with AES-128 encryption, which is outdated and vulnerable.
- Their watermarking techniques are basic, embedding simple logos , which are susceptible to tampering and lack advanced protection.
- Some systems use pixel intensity modifications for watermarking but fail to incorporate traceable identification mechanisms for data leakage.
- There is a lack of comprehensive authentication and malicious activity detection frameworks, making these systems inadequate for modern cloud security challenges.

Existing System

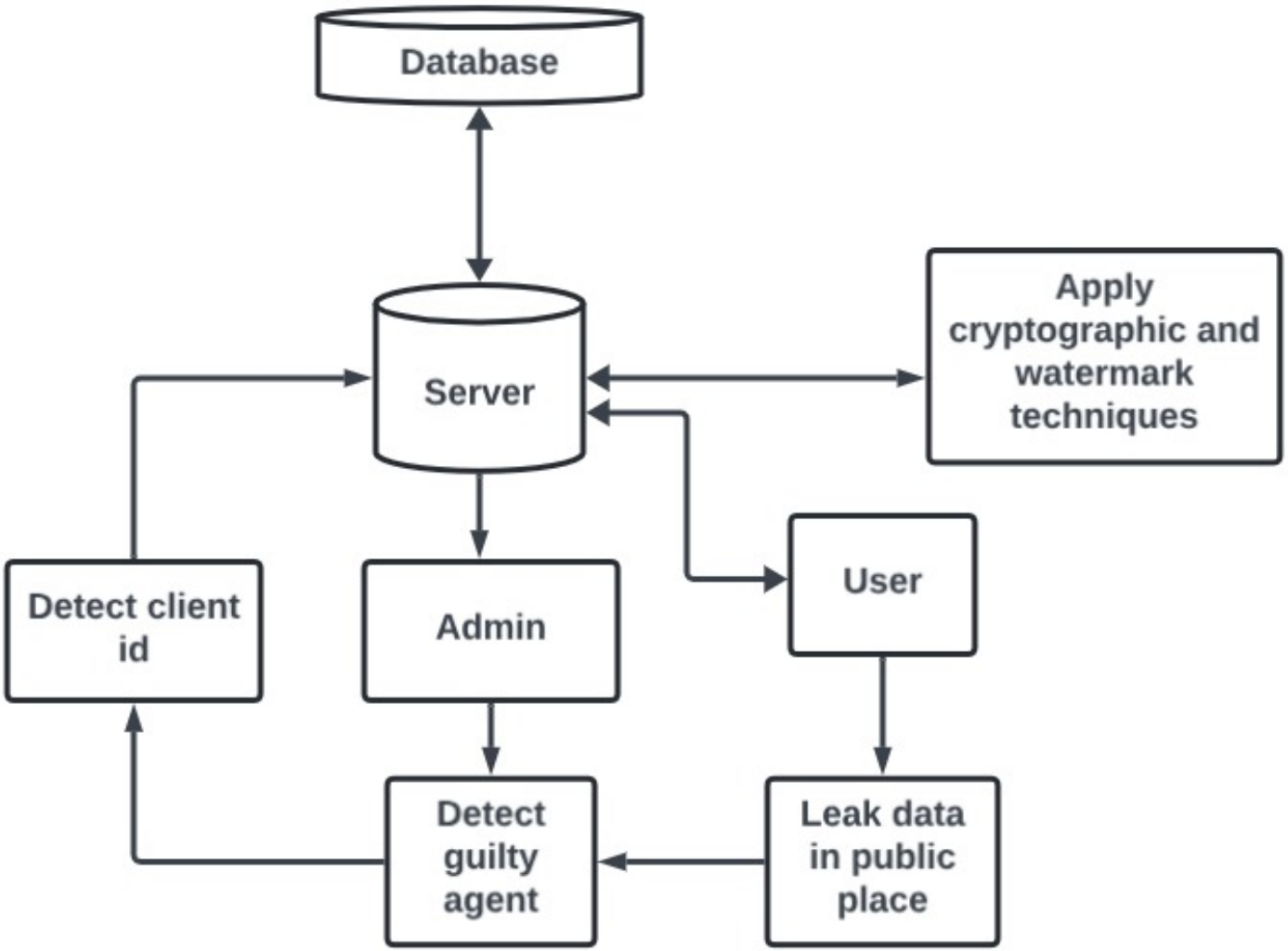

Fig 1: Existing System Architecture

Proposed System

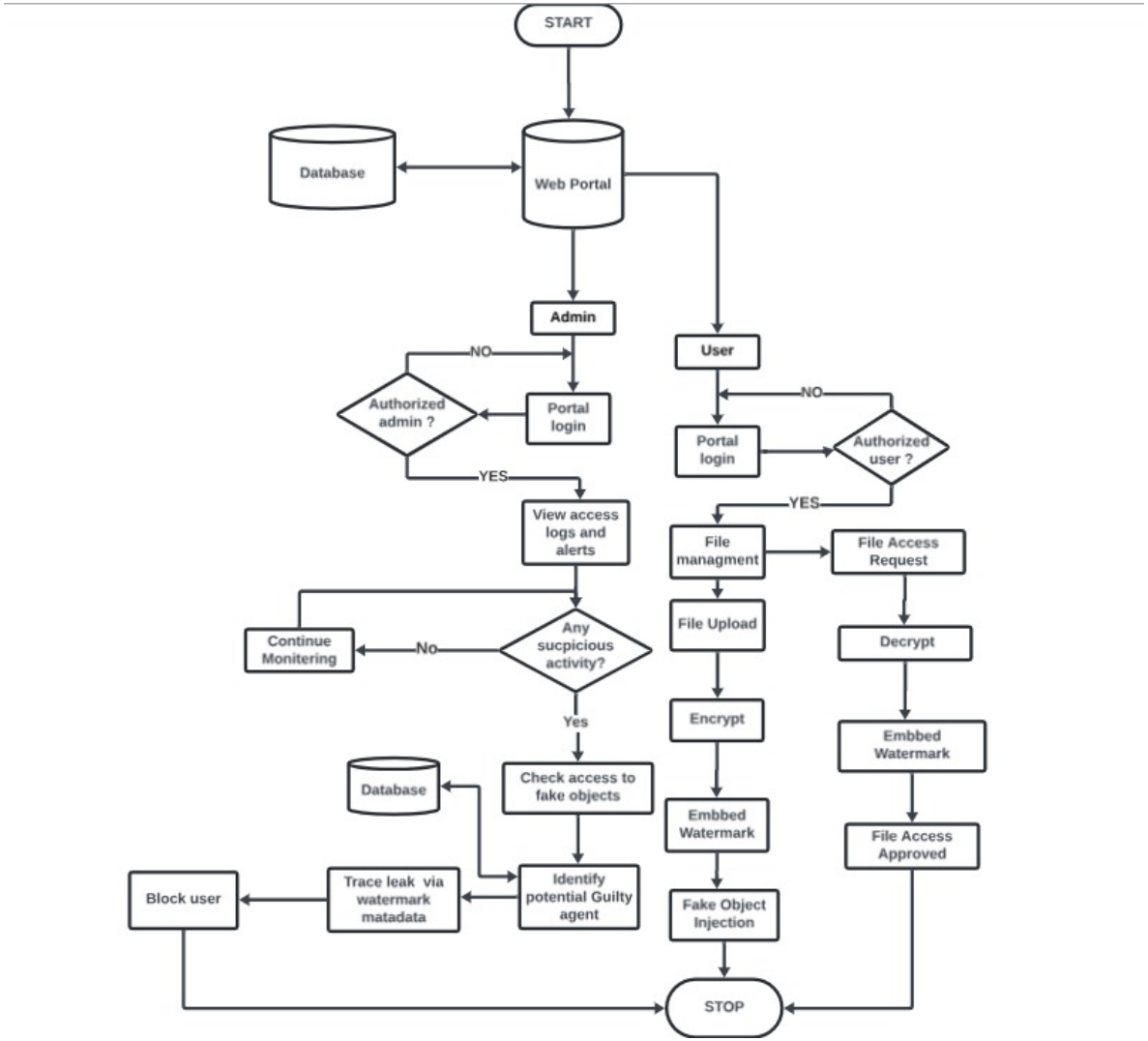

Fig 2: Proposed System Overview

Proposed System

- **File Encryption:** AES-256 Encryption: Securely encrypts files upon upload for safe cloud storage.
- **Watermarking:** Unique Watermarks: Embedded each time a user accesses a file, associating metadata with the user. Secure Database: Tracks user identity and access time for traceability.
- **Decoy Injection:** Fake Objects: Inserted as decoys to detect unauthorized access attempts.
- **Access Verification:** Checks user authorization and monitors actions through decryption and watermarking. Behavior Analysis: Continuously analyzes access logs for suspicious behavior and alerts the admin.
- **Admin Response:** Investigation: Review logs, verify interactions with fake objects, and trace leaks via watermark metadata. Action: Block access or alert users to prevent further leaks, enhancing overall cloud security.

Algorithm

AES

-**Input:** Plaintext file P , 256-bit key K
-**Key Expansion:** Generate round keys from KKK.
-**Initial Round:** AddRoundKey: XOR PPP with the first round key.
-**Main Rounds (14 rounds for AES-256):**
SubBytes, ShiftRows, MixColumns (except final round), AddRoundKey

Algorithm

Watermarking

-Convert Identifier to Binary and Format Binary Data
-Embed Watermark into File
-Extract and Interpret Watermark
-Decode Binary Data to Identifier

Fake object Injection

-First Define Fake Object
-Generate and Insert Fake Object
-Monitor Access and Analyze Interactions
-Take Action

Hardware and Software Requirements

Hardware :

- Operating System : Intel Core i3 or higher
- Speed: 2.4 GHz or faster
- RAM: 4 GB (minimum), 8 GB or higher recommended
- Hard disk : Minimum 250gb

Software :

- Operating System : Windows 10 or higher
- Application Server : XAMPP, Apache
- Frontend : HTML, CSS, JavaScript
- Backend : PHP
- Database : MySQL

Future Scope

Advanced Technologies Integration:

- Integrate AI and machine learning to detect sophisticated threats and anomalies.
- Implement real-time analytics and automated responses for proactive threat management.

Broader Platform Compatibility:

- Expand the system's applicability across different cloud platforms and storage solutions.

Performance Optimization:

- Optimize encryption and watermarking processes to improve efficiency and maintain high security standards.

Conclusion

In this phase, we successfully integrated the watermarking algorithm with the portal system, allowing for seamless embedding of unique metadata within the files uploaded through the portal. This integration enhances the security of the system by ensuring traceability and detecting unauthorized access. Meanwhile, the other algorithms continue to run independently, contributing additional layers of security. With this progress, we have strengthened the foundation of our system, and in the next phase, we aim to integrate all algorithms into a unified system, improving both its functionality and security.

Thank You!