DATA LEAKAGE DETECTION IN CLOUD COMPUTING

Project BlackBook
Submitted in Partial Fulfillment of the Requirements
For the Degree of

BACHELOR OF TECHNOLOGY
BY

Akshat Gandhi - 1514127

Bhavna Varshney - 1624010

Anurag Varshney - 1624011

Under the Guidance of
Prof Mrs. SANGEETA NAGPURE

DEPARTMENT OF INFORMATION TECHNOLOGY
K.J.Somaiya College of Engineering,Mumbai - 77
(Autonomous College Affiliated to University of Mumbai)
2018-2019

Abstract

With the advent of the Modern Era of Technology, sprung the dependence of an individual on the availabilility and flexibility of the Internet. Today the internet is used not only for communication but also for correspondence and as a secondary shared system for information exchange, accessible to those who have the permission to do so. However with the advent of this migration to internet sharing, came the need for online storage - the cloud. This made the information readily available to a multitude of users, but at the same time vulnerable to breaches by unauthorized individuals. This system aims to provide a way to securely store shared data in the cloud while also allowing for trace ability of the same in the event of a breach.The system will encrypt the data prior to storage, allow decrypted data access only to authorized users and backtrack any guilt agent in the event of a leakage.

Keywords: Data leakage, Agent, Backtracking, String manipulation, Meta data manipulation.

Contents

List of Figures

List of Tables

1 Introduction

This chapter discusses the problem statement and the proposed solution.

1.1 Problem Definition

Currently deployed cloud computing solutions do not provide a secure cloud storage for the end users.
The solutions do not provide the end user with any method of self encrypting the files on the cloud, the user can encrypt files separately and upload but the cloud itself does not encrypt files in a manner that the files will only be accessible to the user. They also do not provide a way to confirm the leakage of data.

1.2 Motivation

The most traditional method of Bemusement or Perturbation is using a watermark. This involves embedding a unique code into the copy of data before it is handed out to the mediator.
The disadvantages to this approach is that it cannot detect the source of leakage.

1.3 Scope

Using a web portal to access the stored user files, encrypting the user files before storage and decrypting during access. Using a discrete method to embed a key in the file for the purpose of tracking the guilt agent.

1.4 Salient Contribution

By using embedded keys, encryption and decryption for storage and access we can mitigate the problem arising due to unsecured access. Utilizing the embedded data we can track the agent that leaked the data.

1.5 Organization of Synopsis

The synopsis consists of project abstract followed by:

- Introduction: It consists of problem definition , motivation and scope of the project.
- Literature Survey: It describes the previous research papers in the field of leakage detection and techniques used by them.
- Software Project Management Plan: This document talks about project deliverables, project organization followed by roles and responsibilities.
- Software Requirement Specification: It portrays the functional and non-functional requirements of the system.

- Software Design Document: It describes the overview of the design also contains the requirement tracability matrix and the architecture of the system.
- Software Test document which contains the test approach and the test cases.
- Lastly the conclusion which describes the modules implemented and the deliverable submitted.

2 Literature

This chapter discusses various techniques that can be used to for Lip Reading.It shows a detailed compare and contrast table of all the techniques mentioned.

2.1 Abstract

With the advent of the Modern Era of Technology, sprung the dependence of an individual on the availabilility and flexibility of the Internet. Today the internet is used not only for communication but also for correspondence and as a secondary shared system for information exchange, accessible to those who have the permission to do so. However with the advent of this migration to internet sharing, came the need for online storage - the cloud. This made the information readily available to a multitude of users, but at the same time vulnerable to breaches by unauthorized individuals. This system aims to provide a way to securely store shared data in the cloud while also allowing for trace ability of the same in the event of a breach.The system will encrypt the data prior to storage, allow decrypted data access only to authorized users and backtrack any guilt agent in the event of a leakage.
Papers Referred:

- Panagiotis Papadimitriou & Garcia-Molina,"Data Leakage Detection", IEEE Transactions on Knowledge & Data Engineering, VOL.23,NO.1,page 51,January 2011

  This paper talks about dealing with data leakage detection using Watermarking techniques. This method a distinguishable code is incorporated within each distributed set. And so tracing a leaker is an easy job if a copy is found with an unauthorized agent. This technique is not full proof as watermarks can be corrupted and partially destroyed, Moreover these attacks are categorized under silent attacks, where knowledge is leaked without any prior knowledge of it .

  In the second section, author establishes Invisible watermarking techniques as a counter for safeguarding sensitive data. Invisible watermarking incorporates a invisible watermark into the image. This technique targets the most prominent section of the data and the incorporation of invisible watermark is such that it can't be separated from the data without degrading the quality of the source image .

  The third section talks about introducing a new classification model. there is a contrast sketching between DLP classification model and it's validity was checked under various constraints. In the end the paper concludes with positive outcomes in support of the model.
  The author also introduces time stamping. Time stamping refers to collaborating time along with the data, this time is maintained by the

computer. The paper talks about various phases, primarily the Learning phase which describes how data is trained to incorporated time stamp with the sensitive data. Later the paper talks about Detection Phase, which is primarily the testing phase here the document is tested against the earlier recorded time stamps in the learning phase, along with a confidential score. The system knows if the document is sensitive or not by comparing the time stamp, if time stamp is bigger or equal to time stamp then document is blocked .
This approach detects the Agent who has supposedly leaked the data. Here a probability factor is calculated on this basis of the agent who has more probability of leaking the data is identified. The probability basically depicts the chances of how likely is it that the agent can be guilty. For this to be done the demand of the system is, a rough calculation of the probability for which the value is needed to be speculated.

The basic aim of all allocation strategies is finding out the source of leakage. The methods discussed so far made no changes to the available data and often ended up inserting unreal objects or datasets for easing the process of finding the guilty agent. This paper talks about efficient distribution ways, it's focus is on ordering techniques to ease the process of detection by smartly distributing the data to the agents.

By far the aim remains the same, detecting the guilty agent. Few techniques propose injection of fake objects during distribution as per the request arising by the agent. The paper tries to identify the exact time as well as the guilty agent by making use of data allocation methods. This paper talks about how identification can be prime lined in the initial distribution phase by the distributor by a simple tactic of injecting fake objects. These injected objects have no correspondence with the actual data but give a appeal of real data to the distributed agent. The concept of embedding watermark is synonymous to this concept. Where similarities can be drawn on the basis that object insertion acts in a similar fashion to hiding watermark. with the help of these fake objects the distributor can easily identify for sure the agent who is responsible for this irresponsibility. This technique also provides evidential proofs to sideline the guilty agent with accuracy.

This paper talks about incorporating data leakage detection with Integrity preservation mining. The paper provides us with various techniques to combat the issue. Beginning with discussing algorithms which will bring about efficient distribution which will in turn help in identifying who leaked the data. The author uses data streaming model. the streaming models facilitate easy computation of association rules.

- Deepthi Rao,Siva Kumar & P.Santhi,”An Efficient Multi User Search able Encryption Scheme without Query Transformation over Outsourced Encrypted Data”,
  New Technologies, Mobility and Security, 2018 9th IFIP International Conference,2018

Cloud computing has become a popular buzzword and come out to be of great success in recent years with many advanced contributions. As cloud is storing an enormous amount of digital data engaged with third party services over the internet which raises new security concerns. Efforts are being made by researchers to make cloud secure and reliable computing environment. The technique used in this paper can be recognized as one of the leading methods to secure our sensitive data. The data we used is weather forecasting data which we have accumulated from the website of the government of India. Proposed methodology balances the load of whole data into chunks so that parallel processing will increase and execution time will decrease.

Also, when any leakage of data comes to our concern identification of the guilty agent is performed. With the help of s-max algorithm we can conclude that it gives a significant improvement to find a guilty agent in probability with respect to ¿ 0.4 of the reduced data. The level of security is computed or analyzed in the range of 0 to 1, with some probability criteria.

3 Software Project Management Plan

The aim of this document is to gather , analyze and give an in-depth insight of complete project.
This document specifies:

- The objective of project
- The project deliverables and milestones
- The roles and responsibilities of the members of the project
- The project dependencies and constraints
- The project timetable

3.1 Introduction

3.1.1 Project Overview

The main purpose of these project is to provide a secure access to the stored data for the authorized users via the web portal. Using of encryption and decryption to securely protect data from unauthorised access and using embedded data to find the leakage agents.

We provide a portal which can be used to encrypt files before they are stored on a cloud server.By maintaining access data, we track the user and the file being accessed we can simplify the process of backtracking the data in the event of a data leak.The portal is how the user will login and encrypt the files before they are uploaded to the cloud storage.The user can also access previously stored files which are then decrypted through the portal before they can be accessed by the user.

The user can give access to another user who will have to login through the portal and then access the said file. The system also preserves the id for all file access.

When a file is accessed, the user id is stored in the file. This makes it easier for the algorithm to find the user who caused the leak by comparing the value embedded in the file with the user data.

3.1.2 Project Deliverables

The project consist of many smaller deliverable module and delivery of those module is essential on time.
Here is the list of all deliverable:

1. Source Code: The delivery of these is estimated on 10 March 2019.
2. Library file: The approximate delivery date is 12 March 2019
3. Document: All the necessary document need to be provide during installation.

3.2 Project Organization

3.2.1 Software Process Model : Component based model

1. The Project team is meeting once a week to discuss the progress made by each member and to share the relevant information and be documents that have been prepared. The number of meetings may increase during the final semester as the team members will have more time.
2. There are reviews being conducted once a week during the team meetings. A complete technical review will be conducted at the end of the Design Phase. There will be reviews conducted at the completion of every testing phase.
3. The major milestones to be achieved are as follows:
    - Results of research of existing system and discussions with the Project leader.
    - Results of interview with experts and team meetings to finalize the requirements of the software.
    - Results of the Design Phase, which include a number of modeling diagrams, like the use cases, class diagrams, etc.
    - Results of the first coding phase will be an initial code that will be then tested.
    - Based on the results of the testing, they code will be reviewed in the second coding phase.

3.2.2 Roles And Responsibilities

We consist the team of 3 members. We divided the responsibility based on the familiarity, schedule and expertise of the member.

3.2.3 Tools and Techniques

- Front end using html,css and bootstrap.
- Using php,nodeJS for scripting and backend.

3.3 Project Management Plan

Tasks

The following tasks are to be executed:-
1. Requirement Analysis Phase 1
2. Requirement Analysis Phase 2
3. Design of System
4. Coding Phase 1
5. Coding Phase 2
6. Testing Phase 1

Requirement analysis:
1. Requirement Analysis Phase 1: This will include the research of existing software and a discussion with the Project guide.

2. Requirement Analysis Phase 2: Based on the above results, the project team will discuss and finalize the requirements that are to be provided. We shall consult a number of experts during this phase. The SPMP shall also be prepared during this phase.

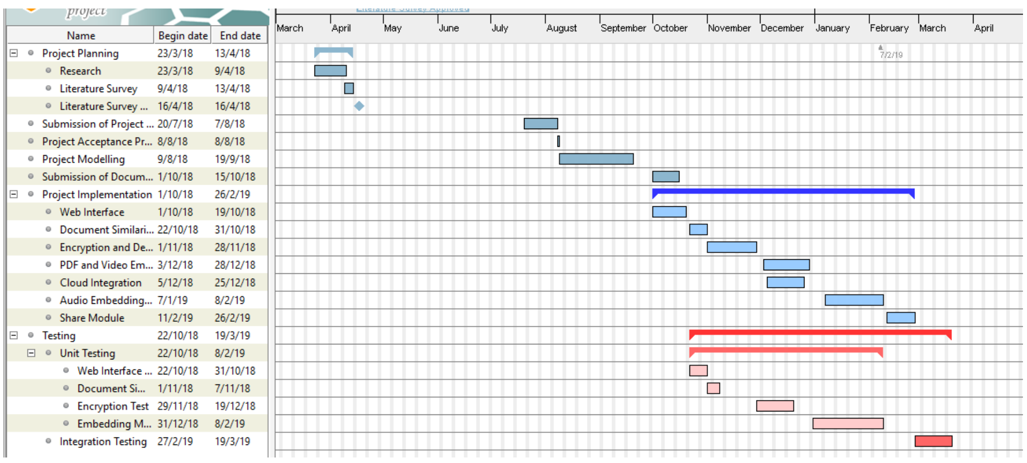
3. Design Phase: The design phase will involve the design of the static view, dynamic view, and the functional view of the software. A number of diagrams including the Use case, class diagram, activity diagram, and data flow diagrams will be used to model the software. Also, the GUIs will be designed during this phase

4. Coding Phase 1: The prerequisite to this phase is the study of Algorithm. After this study, an initial code of the entire project will be written. Also, the database will be created during this phase. Finally, we shall conduct unit tests.

5. Coding Phase 2: This phase will include a review of the code created in Phase 1. After the review, the necessary code and database will be modified to include the results of review.

6. Testing Phase: We shall be following a testing program that will involve unit testing, integration testing, and validation testing.

3.4 TimeLine Chart



Name	Begin date	End date
Project Planning	23/3/18	13/4/18
Research	23/3/18	9/4/18
Literature Survey	9/4/18	13/4/18
Literature Survey ...	16/4/18	16/4/18
Submission of Project ...	20/7/18	7/8/18
Project Acceptance Pr...	8/8/18	8/8/18
Project Modelling	9/8/18	19/9/18
Submission of Docum...	1/10/18	15/10/18
Project Implementation	1/10/18	26/2/19
Web Interface	1/10/18	19/10/18
Document Similari...	22/10/18	31/10/18
Encryption and De...	1/11/18	28/11/18
PDF and Video Em...	3/12/18	28/12/18
Cloud Integration	5/12/18	25/12/18
Audio Embedding...	7/1/19	8/2/19
Share Module	11/2/19	26/2/19
Testing	22/10/18	19/3/19
Unit Testing	22/10/18	8/2/19
Web Interface ...	22/10/18	31/10/18
Document Si...	1/11/18	7/11/18
Encryption Test	29/11/18	19/12/18
Embedding M...	31/12/18	8/2/19
Integration Testing	27/2/19	19/3/19

4 Software Requirement Specification

The SRS is the document which gives complete description about how the system is expected to perform.It contains the requirement specifications that answer what exactly project is about and what are the needs to complete the project such as functional and non-functional requirements

4.1 Introduction

4.1.1 Problem Definition

The main purpose of these project is to provide a secure access to the stored data for the authorized users via the web portal. Using of encryption and decryption to securely protect data from unauthorised access and using embedded data to find the leakage agents.

4.1.2 Document Convention

These document follows a standard IEEE formatting guideline

- Font style: Times New Roman
- Font size: 12 or 14 for text, 16 or 18 for headings
- Line spacing is maintained at 1em and single text spacing is used

4.1.3 Intended Audience and reading suggestions

This software is intended to be used in conjunction with an existing IT based organizational system to provide additional security. Thus it can be used in any organizational setting where data files are shared among internal users and sensitive data is used.

4.1.4 Project Scope

We provide a portal which can be used to encrypt files before they are stored on a cloud server.By maintaining access data, we track the user and the file being accessed we can simplify the process of backtracking the data in the event of a data leak.The portal is how the user will login and encrypt the files before they are uploaded to the cloud storage.The user can also access previously stored files which are then decrypted through the portal before they can be accessed by the user.

The user can give access to another user who will have to login through the portal and then access the said file. The system also preserves the id for all file access.

When a file is accessed, the user id is stored in the file. This makes it easier for the algorithm to find the user who caused the leak by comparing the value embedded in the file with the user data.

4.1.5 References

IEEE papers:

- Panagiotis Papadimitriou & Garcia-Molina,"Data Leakage Detection", IEEE Transactions on Knowledge & Data Engineering, VOL.23,NO.1,page 51,January 2011

- Deepthi Rao,Siva Kumar & P.Santhi,"An Efficient Multi User Search able Encryption Scheme without Query Transformation over Outsourced Encrypted Data",
New Technologies, Mobility and Security, 2018 9th IFIP International Conference,2018

4.2 Specific Requirements

4.2.1 Product Perspective

The software to be developed will accept user files, embed them with the user id, encrypt the files and then store the data into the respective cloud storage bucket for the user. It will also decrypt the data, re-embed the new user id into the data and share it to a secondary user.

4.2.2 Product functions

The functionality of the system is as follows:

- Login the user to the portal and connect to the respective bucket on the cloud storage
- Encrypt and decrypt data during transfer of data to and from the cloud server
- Embed data with the user id for every access to the file by any user
- Generate a report with access list of all user for any given file
- Identify Similarity of leaked files with stored data

4.2.3 User Classes and characterstics

4.2.4 Operating Enviroment

Since the software is to be a web based portal it can be run using any operating system - Linux, Windows, Mac OS - provided an active internet access is available. The network and firewall must allow any user to send and recieve data using REST api at a high speed to facilitate data transfer between the user machine and cloud.

4.2.5 Designing and implementation constraints

- The user can only upload a single file at a time
- The file format is limited to pdf, mp3, mp4 and flac files

4.2.6 Assumption and Dependencies

The database for the user account management will be handled by the organization. The project is implemented using php as the backend. Thus we required the use of several different dependencies - composer, ffmpeg, getId3 and aws-php-sdk.

4.2.7 External Interface Requirements

User Interfaces

- Web page to Login for user and Admin
- Uploading of files
- Retrieval and downloading files
- File access and sharing page
- Result page for admin
- Admin function page

Hardware Interfaces

Since we are building a web-based service there are no such Specific Hardware Interfaces required.The server handles almost all the work of the entire project and there is no such dependency on the user end.

Software Interfaces

- Composer
- MySQL

Communications Protocols

Since this project is an integration of several different purpose languages and their interdepency, then communication between the componenets is handled by the web browser or the components themselves.

4.2.8 Software Product Features

Functional Requirements

- ID:FR1 : Login to the portal
- ID:FR2 : Create user for Admin
- ID:FR3 : Validation
- ID:FR4 : File Upload
- ID:FR5 : Encrypt User File
- ID:FR6 : Display Stored Files
- ID:FR7 : Retrieve Files
- ID:FR8 : Decrypt Files
- ID:FR9 : Embed user data
- ID:FR10 : File Similarity for Admin
- ID:FR11 : Report Generation
- ID:FR12 : Log out

4.2.9 Software System Attributes

Reliability
Scale: The capability of the sytem to identify and reject incorrect credentials or key.
Meter: Measurements obtained from 100 test cases
Must: Reject 100 cases for incorrect credentials and key
Plan: Reject 99 cases

Availability
Scale: Average system availability(other than network failure)
Meter: Measurements from 100hrs of use
Must: Be available 24/7
Plan: Available 24/7

Security
Scale: Try access to unauthorised files
Meter: 100 tries
Must: Reject all tries
Used: AES Encryption

Maintainability
Since the system is based on a MVC architecture, any change in one module will not affect other modules and addition of external components can be easily done.

Portability
The system is deployed on our machine as a hosted server, accessed using a web portal via the network hence it can be easily accessed remotely.

4.2.10 Database Requirements

MySQL - Used to store User Credentials and data, requires phpmyadmin configuration.
MongoDB - Used to store User Files, requires NodeJS, GridFS and mangoose to be installed correctly.

4.2.11 Data Flow Diagram

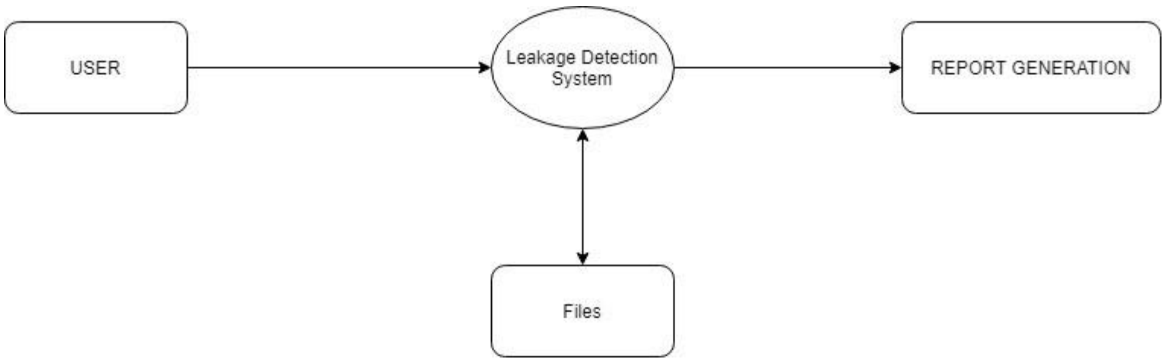


Figure 1: DFD 0

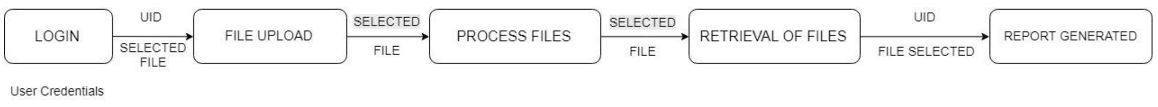


Figure 2: DFD 1

5 Software Design Document

SDD is a representation of the software system to be created.It provides detailed design information needed for planning, analysis, and implementation of the software system

5.1 Introduction

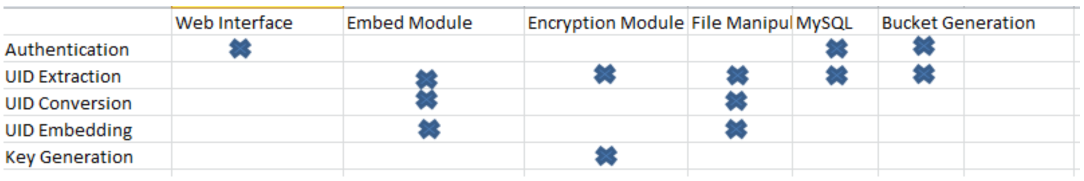
The main purpose of these project is to provide a secure access to the stored data for the authorized users via the web portal. Using of encryption and decryption to securely protect data from unauthorised access and using embedded data to find the leakage agents.

We provide a portal which can be used to encrypt files before they are stored on a cloud server.By maintaining access data, we track the user and the file being accessed we can simplify the process of backtracking the data in the event of a data leak.The portal is how the user will login and encrypt the files before they are uploaded to the cloud storage.The user can also access previously stored files which are then decrypted through the portal before they can be accessed by the user.

The user can give access to another user who will have to login through the portal and then access the said file. The system also preserves the id for all file access.

When a file is accessed, the user id is stored in the file. This makes it easier for the algorithm to find the user who caused the leak by comparing the value embedded in the file with the user data.

5.2 Requirements Traceability Matrix



	Web Interface	Embed Module	Encryption Module	File Manipul	MySQL	Bucket Generation	
Authentication	✖				✖	✖	
UID Extraction		✖	✖	✖	✖	✖	
UID Conversion		✖		✖			
UID Embedding		✖		✖			
Key Generation			✖				

Figure 3: Tracability Matrix

5.3 System Architecture Design

Model View Controller (MVC)

MVC in context of Data leakage detection

1. **Model**
   Model here is nothing but database that is ultimately used for storage for data and if primary data storage goes down one can easily access the data through database.
2. **View**
   View generally include front end login page and pages that gives direct upload and download of file. It also include pages for admin for adding and deleting the users.
3. **Controller**
   Controller generally include various algorithm and test set data for testing purpose. one of the main program is finding guilt agent.

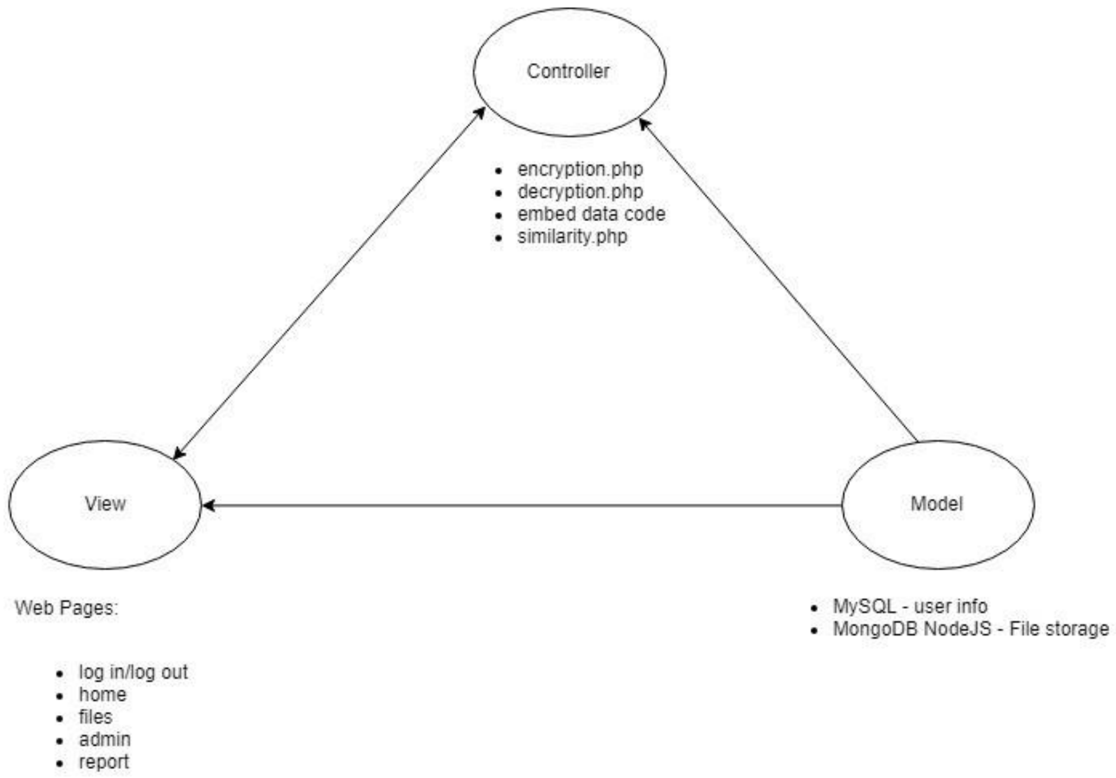
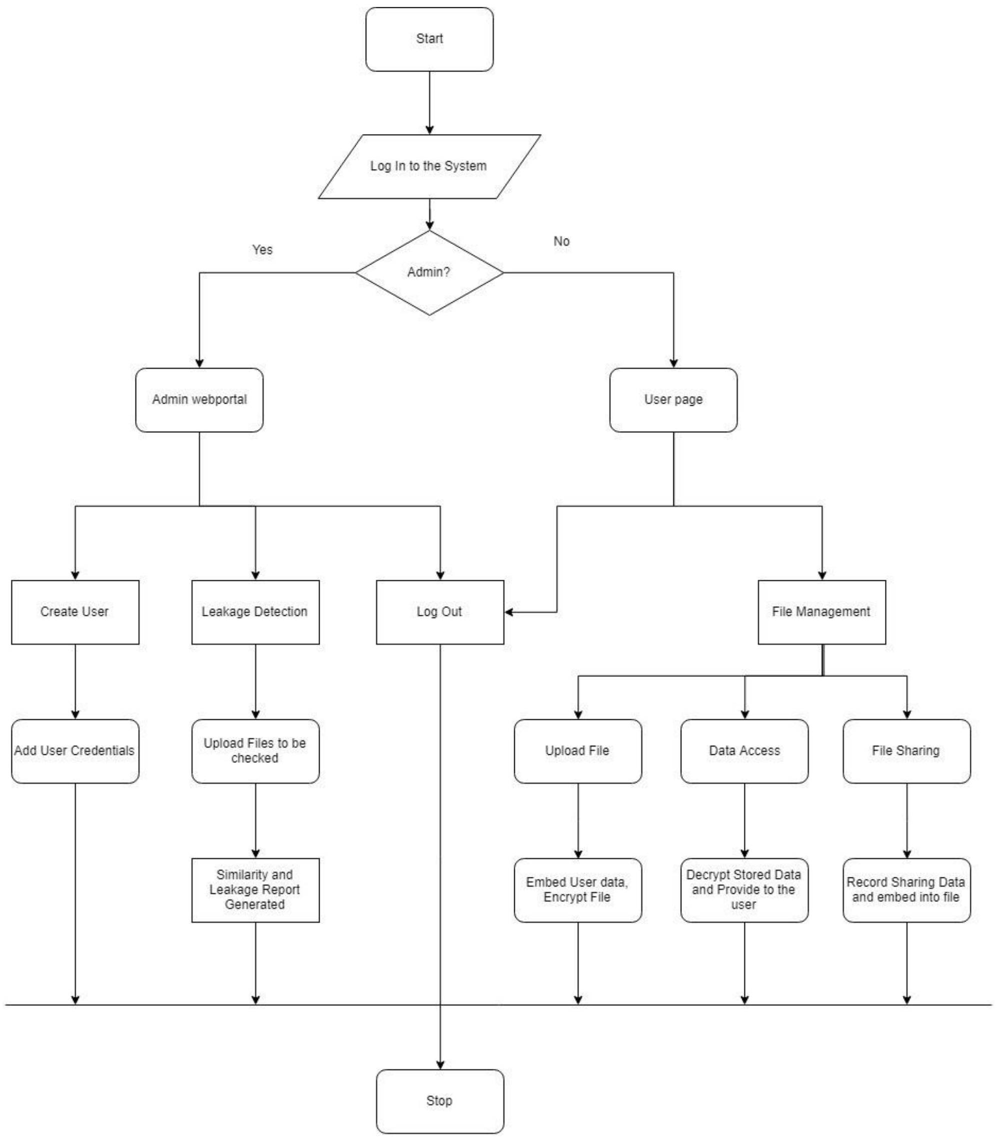


Figure 4: MVC Architecture model


Start
Log In to the System
Admin?
Yes
No
Admin webportal
User page
Create User
Leakage Detection
Log Out
File Management
Add User Credentials
Upload Files to be checked
Similarity and Leakage Report Generated
Upload File
Data Access
File Sharing
Embed User data, Encrypt File
Decrypt Stored Data and Provide to the user
Record Sharing Data and embed into file
Stop

Text

Figure 5: Flowchart of the system

5.4 Component Description

5.4.1 Component Design

Table 3.1 : Component: Web Interface

Responsibilities	1. Give an interface to user, to use the system 2. User is able to upload files which will be stored on the cloud after embedding and encryption
Constraints	1. User should provide valid file format 2. Validation of fields should be follow by the users 3. Supported in modern browsers preferably chrome
Composition	1. Front End: HTML,CSS,Bootstrap
Interactions	1. User upload the file by dragging the file in the upload region or using the upload button
Resources	Web browser

Table 3.2 : Component: Data Encryption

Responsibilities	1. Key generation 2. Sub byte generation 3. Encrypt and decrypt all files handle by the portal
Constraints	1. Source file will be handled by the php layer. The module does not have access to direct path of the file
Composition	1. php and binary libraries
Interactions	1. None
Resources	Valid version of php

Table 3.3 : Component: Cloud Interface

Responsibilities	1. Maintaining buckets for each individual users 2. Using REST api to upload, download, share or delete data
Constraints	1. User files are transmitted using GuzzleHTTPHandler and thus the client machine required a valid cUrl Certificate 2. The network bandwidth must be sufficiently large to accomodate file transfer
Composition	1. Composer, AWS-sdk, php
Interactions	1. None
Resources	Web browser, aws account

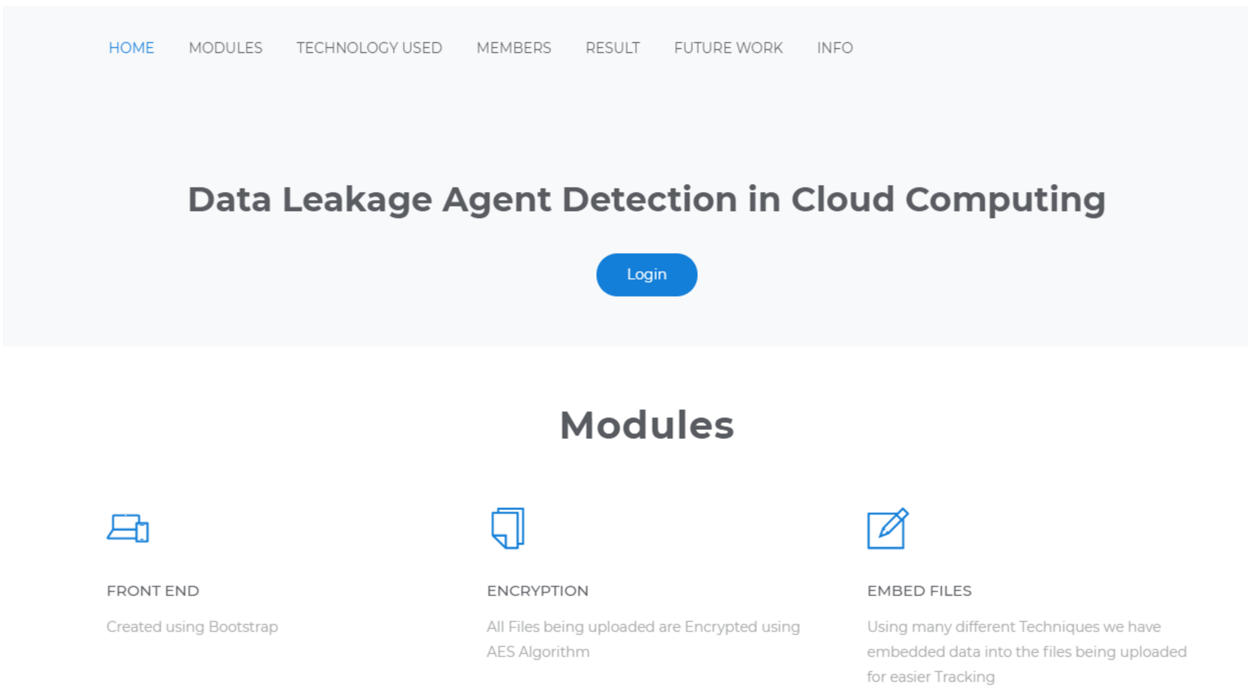
Table 3.4 : Component: Embedding Module

Responsibilities	1. To embed UID into files in a hidden manner
Constraints	1. Data to be embedded should be accomodated without altering the original file 2. Data to be embedded should be accomodated without altering the file size significantly
Composition	1. Php composer
Interactions	1. Cloud module, Encryption and decryption module, php composer
Resources	working php configuration

Table 3.5 : Component: Report Generation

Responsibilities	1. Check uploaded file with another file and determine the similarity between the two 2. Generating a list of access to the files and report generation
Constraints	1. Uploaded file should have atleast 1 access record for report to be generated
Composition	1. php
Interactions	1. Uses the embedding module in conjunction with user database to identify guilt agent
Resources	working php configuration

5.4.2 UI Design



Data Leakage Agent Detection in Cloud Computing

Modules

FRONT END

Created using Bootstrap

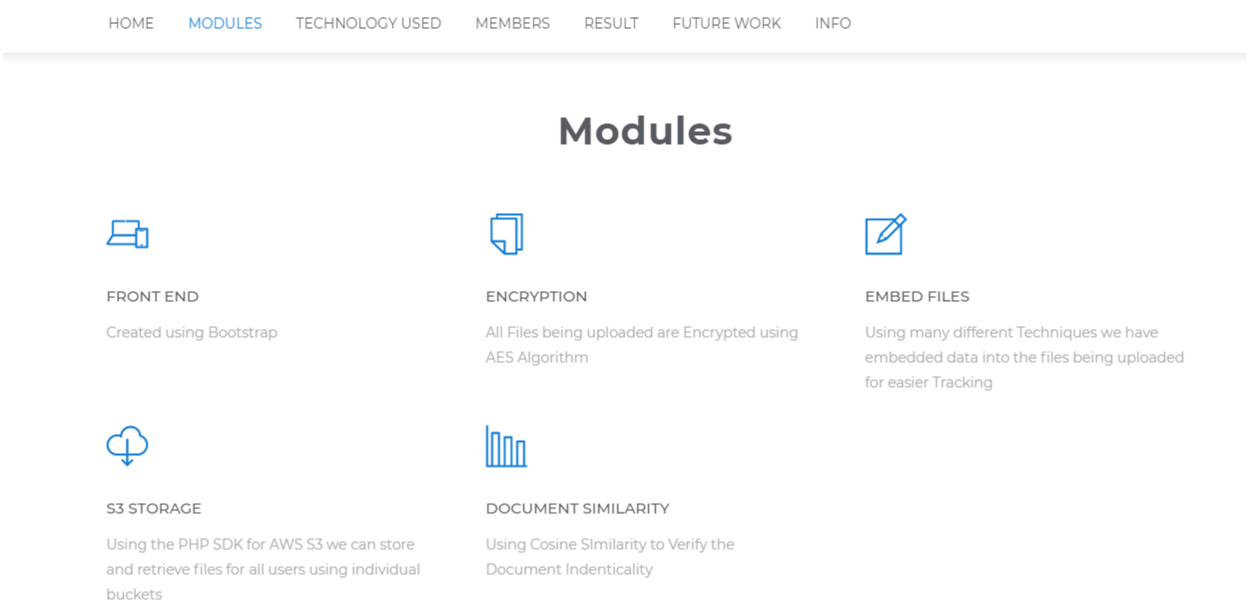
ENCRYPTION

All Files being uploaded are Encrypted using AES Algorithm

EMBED FILES

Using many different Techniques we have embedded data into the files being uploaded for easier Tracking

Home Page



Modules

FRONT END

Created using Bootstrap

ENCRYPTION

All Files being uploaded are Encrypted using AES Algorithm

EMBED FILES

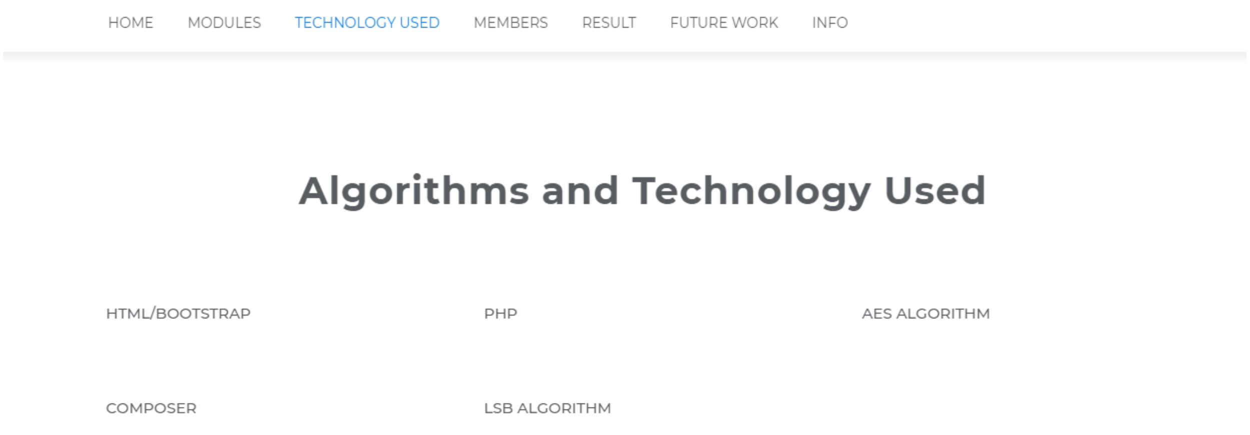
Using many different Techniques we have embedded data into the files being uploaded for easier Tracking

S3 STORAGE

Using the PHP SDK for AWS S3 we can store and retrieve files for all users using individual buckets

DOCUMENT SIMILARITY

Using Cosine SImilarity to Verify the Document Indenticality

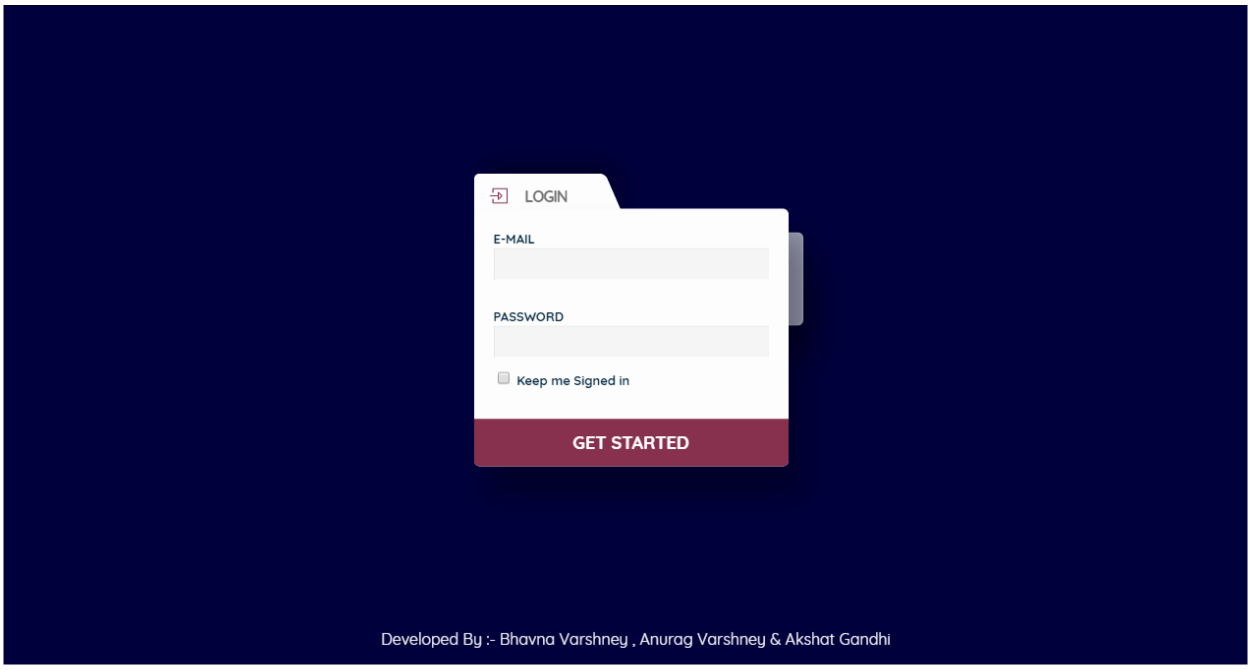

HOME MODULES TECHNOLOGY USED MEMBERS RESULT FUTURE WORK INFO

Algorithms and Technology Used

HTML/BOOTSTRAP PHP AES ALGORITHM

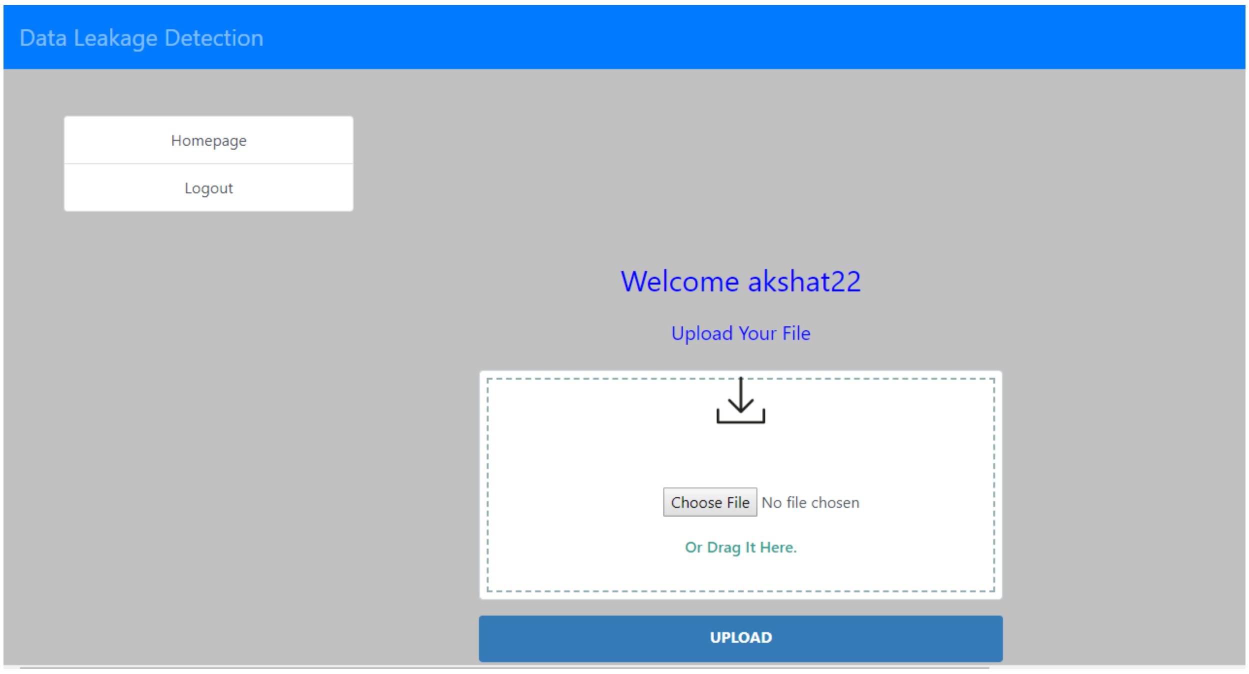
COMPOSER LSB ALGORITHM

Login Page

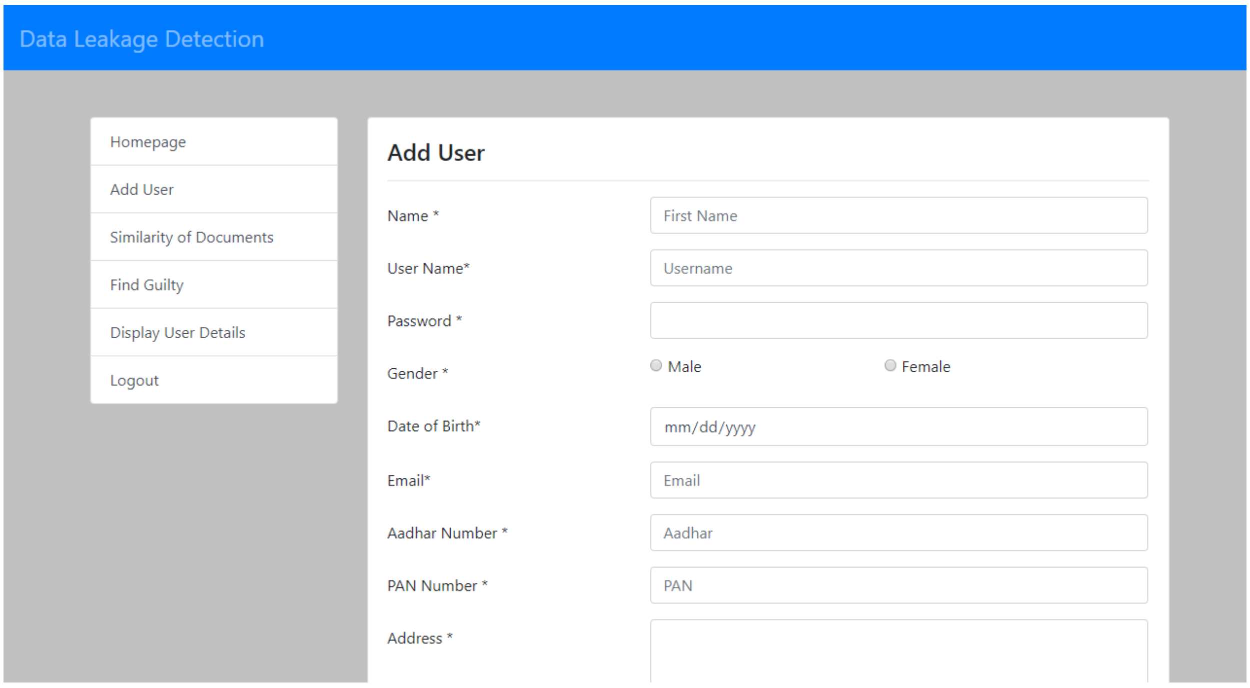


Developed By :- Bhavna Varshney , Anurag Varshney & Akshat Gandhi

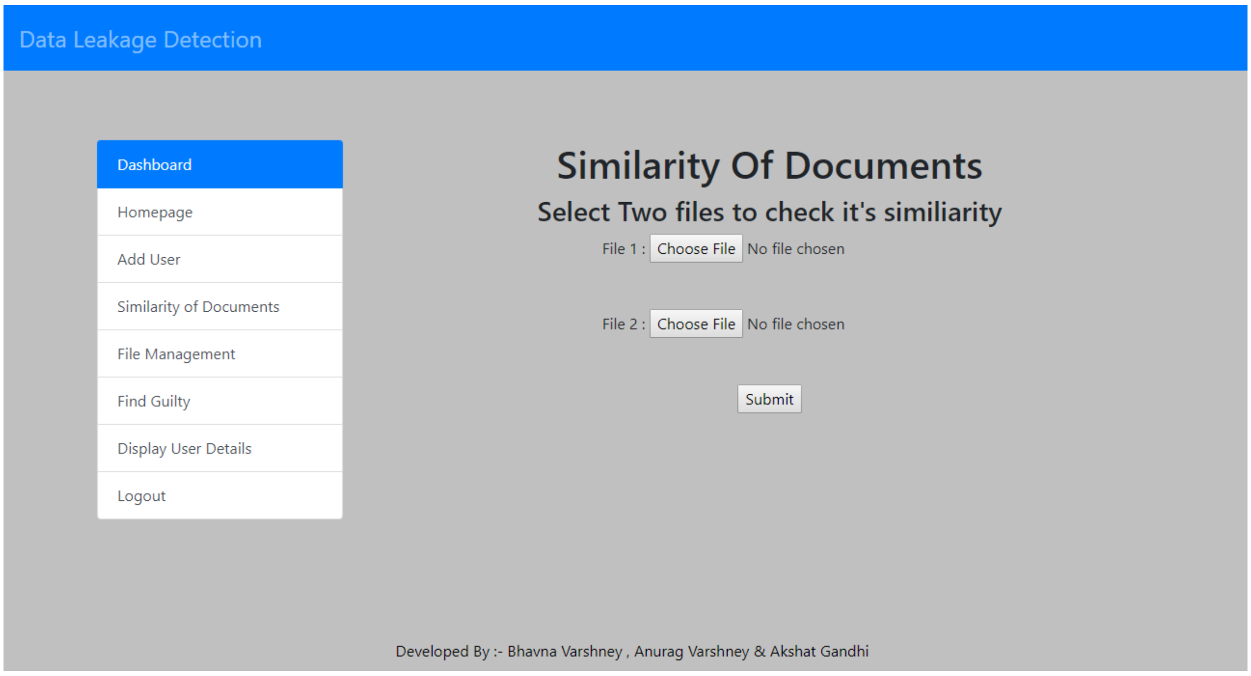
File Management Page



Admin Page



File Similarity



Similarity Of Documents

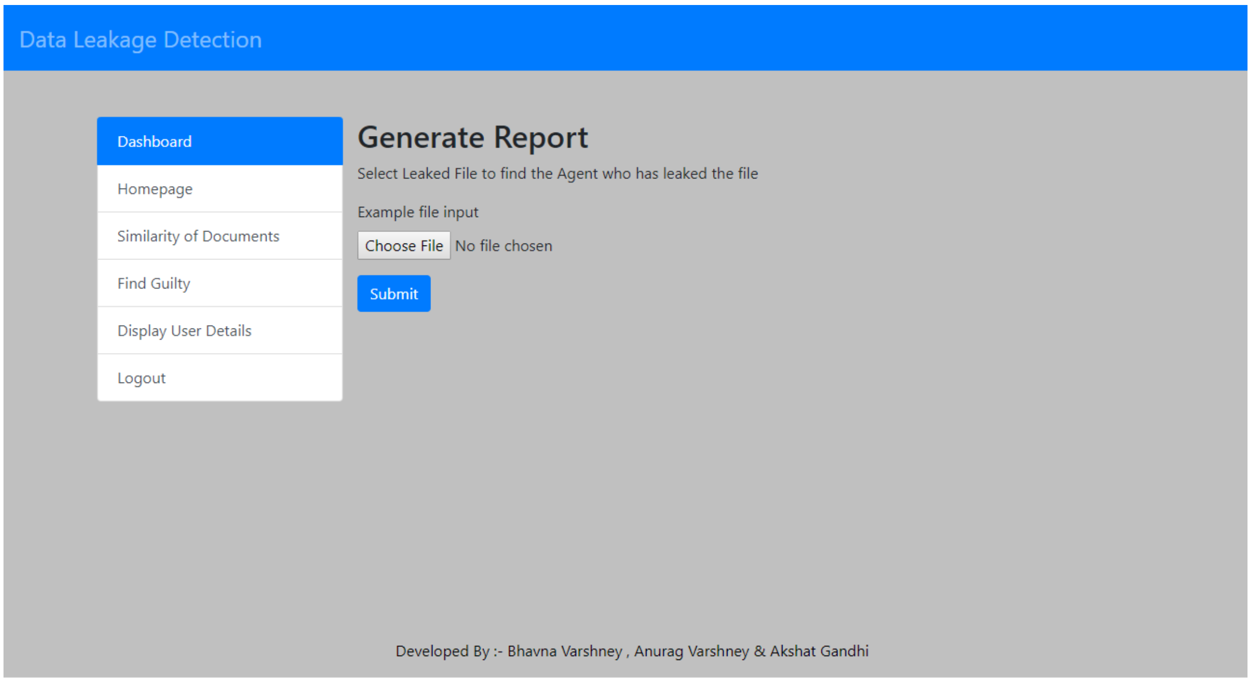
Select Two files to check it's similiarity

File 1 : No file chosen

File 2 : No file chosen

Developed By :- Bhavna Varshney , Anurag Varshney & Akshat Gandhi

User Details Page



Generate Report

Select Leaked File to find the Agent who has leaked the file

Example file input

No file chosen

Developed By :- Bhavna Varshney , Anurag Varshney & Akshat Gandhi

6 Implementation

This chapter provides the details of implementation.

6.1 Pre-processing

6.1.1 Extension Identification

Identify file formats using extensions to determine methods to be used for embedding of and extraction from data.

6.1.2 Data extraction

Extract existing meta data from files to identify ownership of file and access list.

6.1.3 Bucket Abstraction

Creation and assignment of data buckets stored on the cloud to their respective owners at the time of creation and handshake.

6.2 Processing

6.2.1 Embedding

Using user id, we embed files for the purpose of agent detection using backtracking in the event of a data leakage. Depending on the type of file and its format different methods are used - string manipulation for pdf files, meta data manipulation for video files and ID tag manipulation for audio files.

6.2.2 Encryption and Decryption

Using 256 bit AES Encryption we generate a key from the user id and transform it to encrypt and decrypt user files.

6.2.3 Storage

Using amazon AWS S3, files are stored on the cloud contained in their respective bucket based on file owner and access given to a secondary user by the owner.

6.2.4 Data similarity

In the event of a data leakage occurence the outsourced file can be compared to company data using cosine similarity as an indicative measure to verify if the data has truely originated from our company.

6.3 Post-processing

6.3.1 Session management

To facilitate data sharing between parallel code segment and to securely maintain user data, we use sessions to ensure secure availability of this sensitive data through out one access period while also not using cookies which reduces memory usage.

7 Software Test Document

This chapter includes all the test scenarios and the testing approaches to test the working of the project.

7.1 Introduction

7.1.1 Design Overview

The main purpose of these project is to provide a secure access to the stored data for the authorized users via the web portal. Using of encryption and decryption to securely protect data from unauthorised access and using embedded data to find the leakage agents.

We provide a portal which can be used to encrypt files before they are stored on a cloud server.By maintaining access data, we track the user and the file being accessed we can simplify the process of backtracking the data in the event of a data leak.The portal is how the user will login and encrypt the files before they are uploaded to the cloud storage.The user can also access previously stored files which are then decrypted through the portal before they can be accessed by the user.

The user can give access to another user who will have to login through the portal and then access the said file. The system also preserves the id for all file access.

When a file is accessed, the user id is stored in the file. This makes it easier for the algorithm to find the user who caused the leak by comparing the value embedded in the file with the user data.

7.2 Test Plan Implementation

We will be performing white and black box testing for our system. Under these testing methods we will perform the following tests -

- Unit Testing: While coding we perform tests on the taken unit to check for errors - encryption, decryption, similarity.
- Integration Testing: Once unit testing has verified that the module is working properly we combine the modules.We integrate the html pages with the encryption module, the similarity code is used with the report page and file retreival with File System.
- Functional Testing: We check the functions used, the php functions used are encryption, decryption.
- Usability Testing: We use the tool Lighthouse.

7.3 Test Case Design

Sr. No	Feature Tested	Mode of testing	Passing criteria	Expected outcome	Actual outcome	Conclusion
1	Validation for password	Unit Test	Password should be of 8-12 digit with one upper case and special character	Data inserted	Data inserted	Passed
2	Validation for password	Unit Test	Password should be of 8-12 digit with one upper case and special character	Data does not match criteria	Data not inserted	Passed
3	Uploading of file	Unit Test	File uploaded to respective user bucket	File uploaded	File uploaded	Passed
4	Uploading of file	Unit Test	File uploaded to respective user bucket	File not uploaded	File not uploaded	Passed
5	Encryption of file	Unit Test	File encryption using user id	File encrypted	File encrypted	Passed
6	Encryption of file	Unit Test	File encryption using user id	File not encrypted	File not encrypted	Passed
7	Decryption of file	Unit Test	File decryption using user id	File decrypted	File decrypted	Passed
8	Decryption of file	Unit Test	File decryption using user id	File not decrypted	File not decrypted	Passed
9	Cosine Similarity	Unit Test	Accurate percentage of similarity	Similarity percentage	Similarity percentage	Passed
10	Cosine Similarity	Unit Test	Accurate percentage of similarity	Not actual Similarity percentage	Not Similar percentage	Passed
11	Embedding UID	Integration Test	Embedding data to file	Data embedded	Data embedded	Passed
12	Embedding UID	Integration Test	Embedding data to file	Data not embedded	Data not embedded	Passed
13	Report Generation	Integration Test	Report for guilt agent	List of access	List of access	Passed

8 Conclusion and result

This chapter provides the conclusion driven from all the work done on the thesis.It also includes the future scope of the thesis

8.1 Conclusion

we have successfully implemented a portal system which provides us with a secure way to manage confidential files for a closed group or co-orperation. In these paper we have demonstrated the advantages of the proposed system over a traditional cloud storage system. By combining the use of encryption methods, watermarking data, file and remote storage system along with a portal to manage all these functionality we have effectively implemented a system which can accurately detect the guilt agent in the event of the data breach or leakage.

8.2 Result

I DataSet
Text and pdf file are used as a dataset.
II Result
The time required for uploading and downloading the files depend upon the internet bandwidth and may vary.

8.3 Future Scope

The current system follows a reactive approch to the problem of data leakage. It does not take any measure to prevent or reduce the occurance of data leakage. For our future scope we can explore the possibility of including a firewall integrated version of the system thus improving the security of the system while also monitoring the system network. We can also explore the idea of adding support for other file types or providing the admin user with a choice over the technique being used to embed data. The current system uses amazon sdk. Thus it only supports S3 cloud integration. We can also explore the path through which the system can be made into a dynamic web service with support for multiple cloud storage option.

References

[1] Panagiotis Papadimitriou Garcia-Molina,”Data Leakage Detection”,IEEE Transactions on Knowledge Data Engineering, VOL.23,NO.1,page 51,January 2011

[2] Madhavi Suryawanshi P.Sarita Patil,”Avoiding the data leakage and providing privacy to data in Networking”,International Conference on Computing Communication Control and automation (ICCUBEA),August 2016

[3] Panagiotis papadimitriou Garcia-Molina,”Data Leakage Detection”,IEEE Transactions on Knowledge Data Engineering, VOL.23,NO.1,page 51,January 2011

[4] Madhavi Suryawanshi & P.Sarita Patil,”Avoiding the data leakage and providing privacy to data in Networking”,International Conference on Computing Communication Control and automation (ICCUBEA),August 2016

[5] Amazon AWS official documentation for PHP SDK

[6] Package List for composer dependencies and configuration

[7] Official ffmpeg website and documentation

[8] Official ID3 documentation

[9] Govinda.K Divya Joseph – “Dynamic Data Leakage using Guilty Agent Detection over Cloud” Proceedings of the International Conference on Intelligent Sustainable Systems (ICISS 2017)

[10] Fatimah Y, Gary B Wills Andrew Gravell - “Exposing Data Leakage in Data Integration Systems” The 9th International Conference for Internet Technology and Secured Transactions (ICITST-2011)