DATA LEAKAGE DETECTION IN CLOUD COMPUTING

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

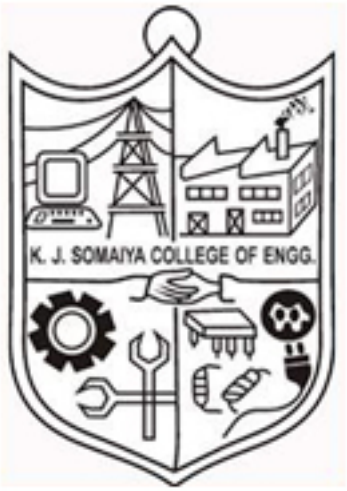
BACHELOR OF TECHNOLOGY
BY

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.

Contents

1 Introduction

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

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:

- Dynamic data leakage detection model based approach for MapReduce computational security in cloud
  Sakshi Chhabra - Department of Computer Applications, NIT Kurukshetra, Haryana, India
  Ashutosh Kumar Singh - Department of Computer Applications, NIT Kurukshetra, Haryana, India
  https://ieeexplore.ieee.org/document/7893234

  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 .

  The second paper, 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 paper 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.
  In this paper the author 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 paper talks detecting 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.

- Dynamic data leakage using guilty agent detection over cloud
  K. Govinda - SCOPE, VIT, Vellore, India
  Divya Joseph - SCOPE, VIT, Vellore, India
  https://ieeexplore.ieee.org/document/8389273

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

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

Task Name	ID	Start Date	End Date
Configuration of Data storage	1	20-Aug-18	20-Sep
Documentation	2	21-Sep-18	10-Oct-18
Front End	3	11-Oct-18	30-Oct-18
Cosin Similarity	4	31-Oct-18	25-Nov-18
Encryption and Decryption	5	31-Oct-18	25-Nov-18
Upload and Retreival Of Data	6	31-Oct-18	05-Dec-18
Evaluation	7	06-Dec-18	08-Dec-18

Figure 1: Time Line Chart

4 Software Requirement Specification

4.1 Introduction

4.1.1 Product 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.

4.1.2 Product 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.

4.1.3 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 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

- NodeJS
- GridFS

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.

4.2.2 Software Product Features

Functional Requirements

- ID:FR1 : Login to the portal
- ID:FR2 : Create user for Admin
- ID:FR3 : Authentication
- ID:FR4 : Choose user file for 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

4.2.3 Software System Attributes

Reliability
Scale: The reliability that system gives during credentials or hash mismatch
Meter: Measurements obtained from 100 test cases
Must: Reject 100Plan: Reject 99Availability
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

Maintainability
Any Changes in the portal structure can be made as long as the user is not signed in. If so the user will log out automatically and has to re-login

Portability
The system is deployed on a cloud server, accessed using a webportal hence it can be easily accessed remotely.

4.2.4 Database Requirements

MySQL/Mongo DB - Requires database to store user credentials.

5 Software Design Document

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 System Architecture Design

Model View Controller (MVC)
MVC in context of Data leakage detection

1. **Model**
   Model here is nothing but hadoop that is ultimately used for storage for data and if primary data storage goes down one can easily access the data through hadoop architecture

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 algoritm and test set data for testing purpose. one of the main program is finding guilt agent.

Flowchart:

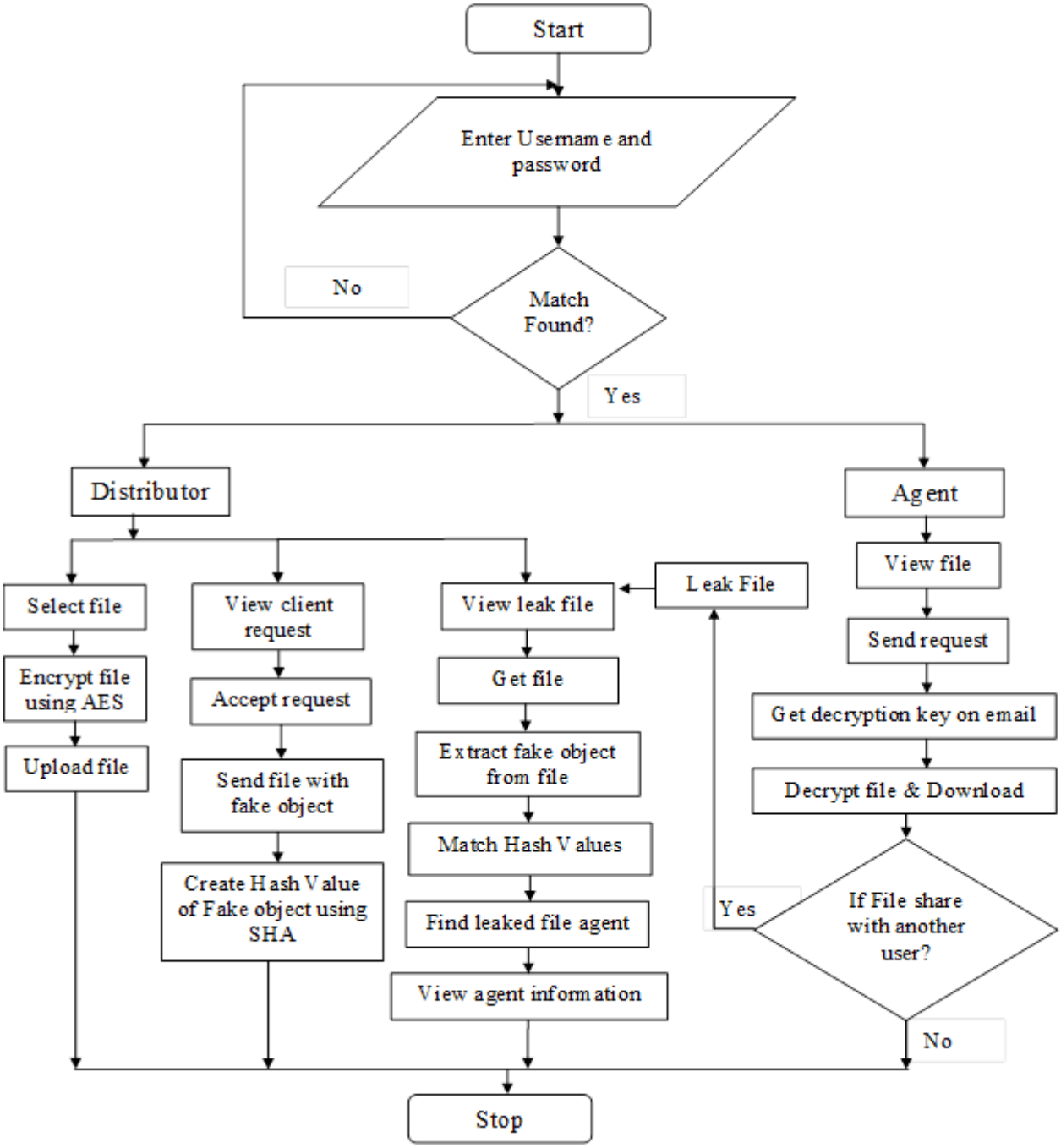


Figure 2: Flowchart of the system

5.3 Component Description

5.3.1 Data Storage Module

(43).png

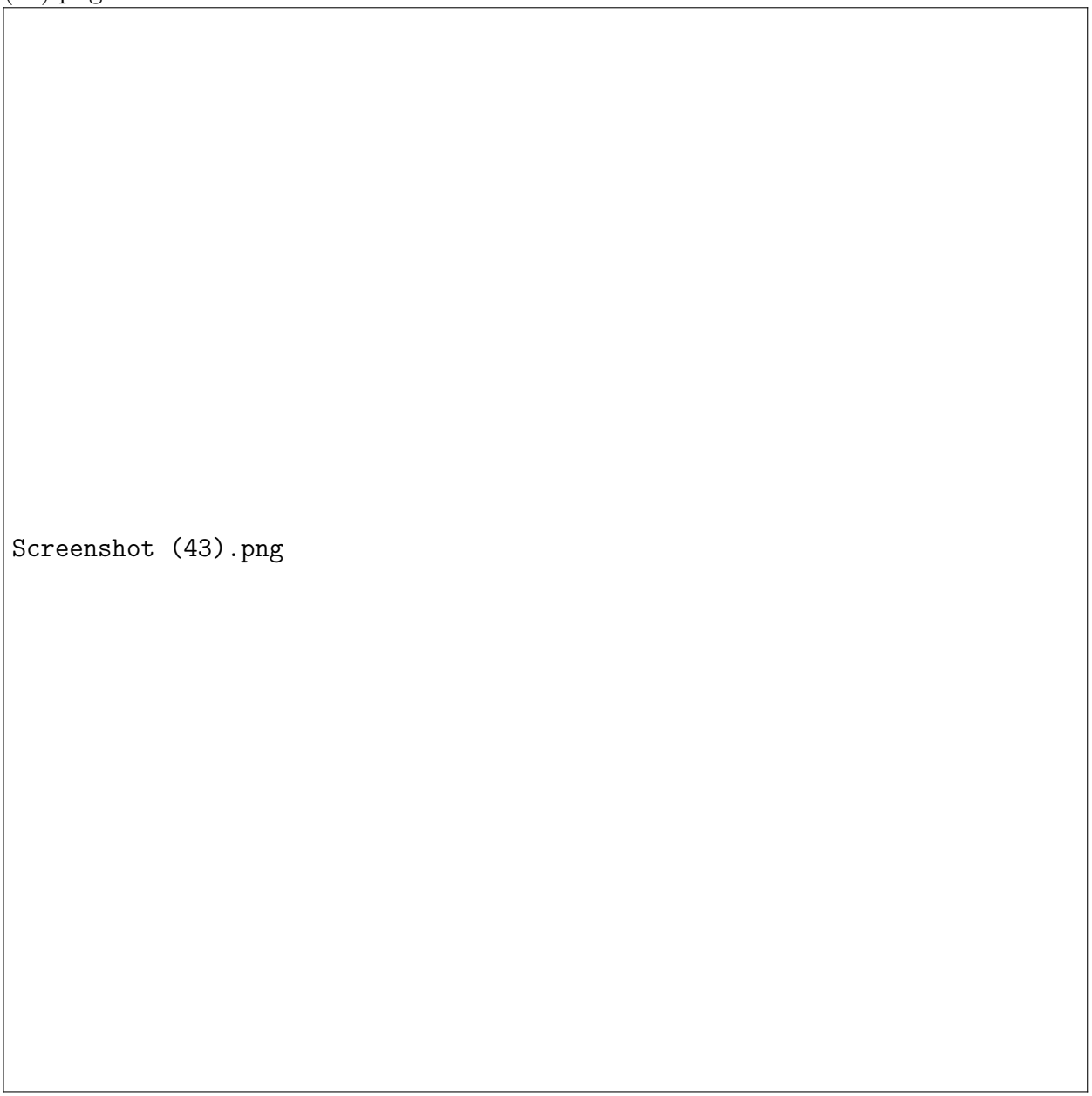


Figure : Store and Retrieve Data to and from the FileSystem.

5.3.2 Security Module

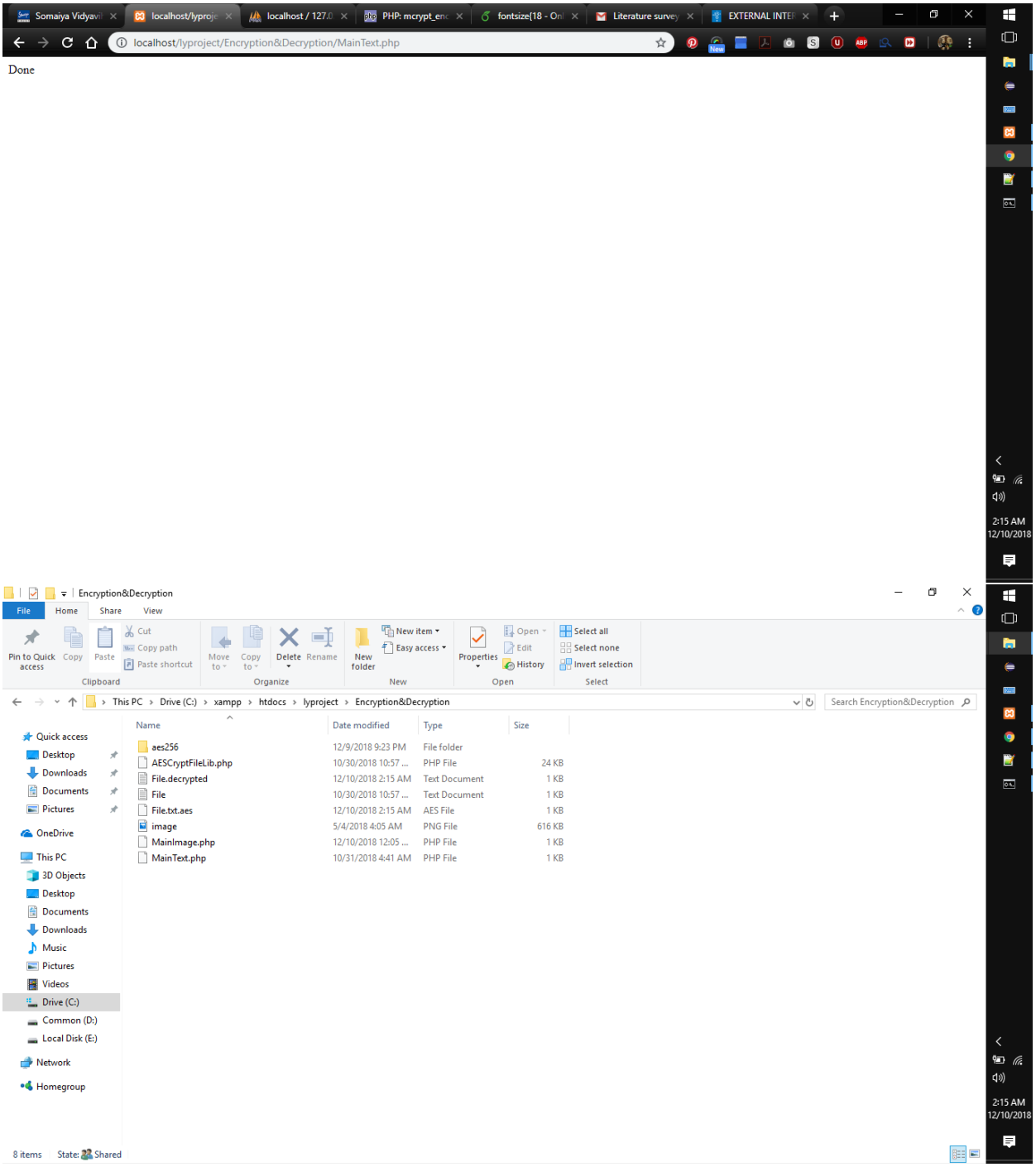


Figure : The Encryption, Decryption of files according to the User ID

5.3.3 Leakage Module

Figure : The embedding of files according to the User ID and access permission, Report generation for document similarity.

5.3.4 User Module

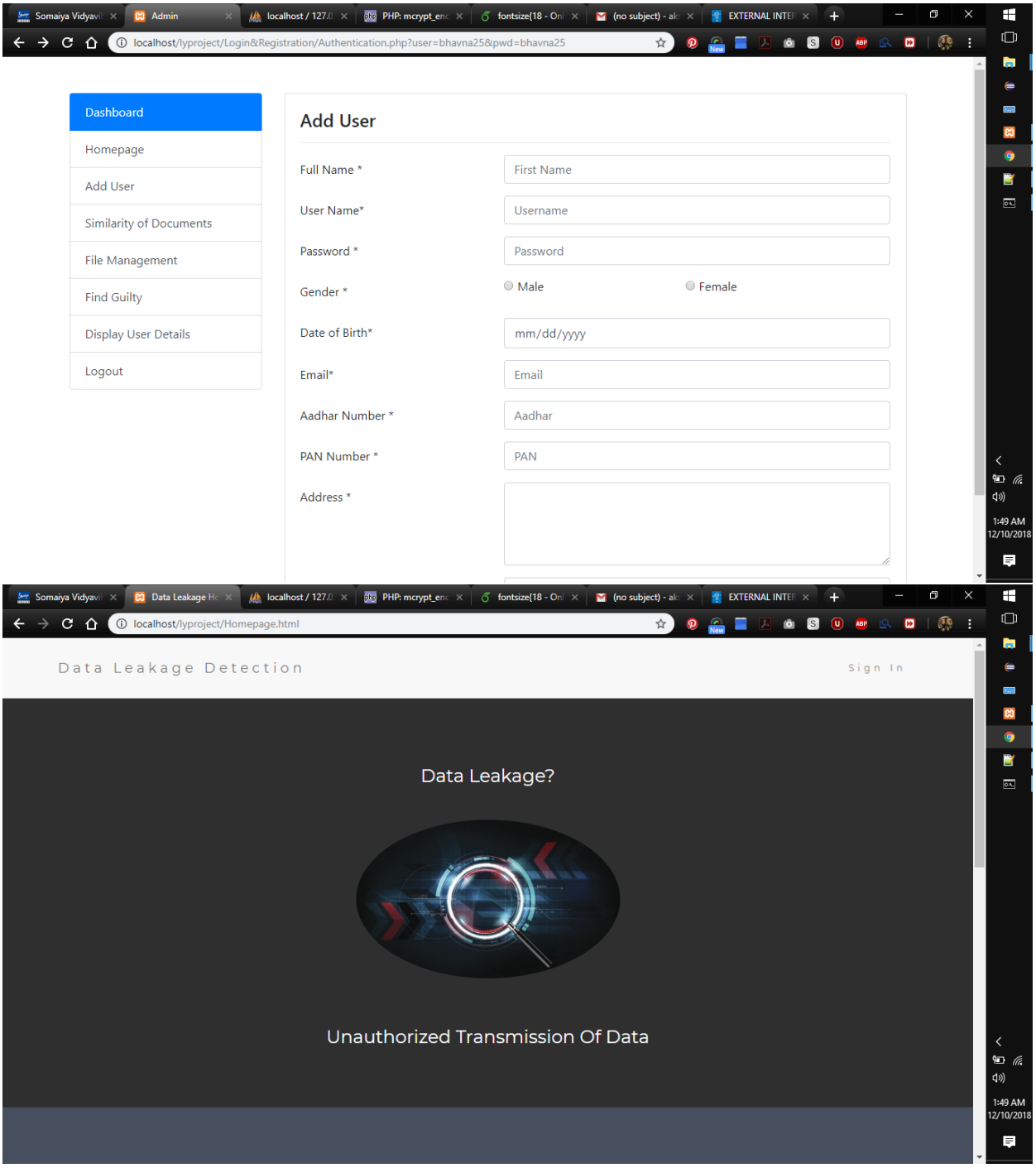


Figure : Create new users.Front end of the Portal.

[width=]

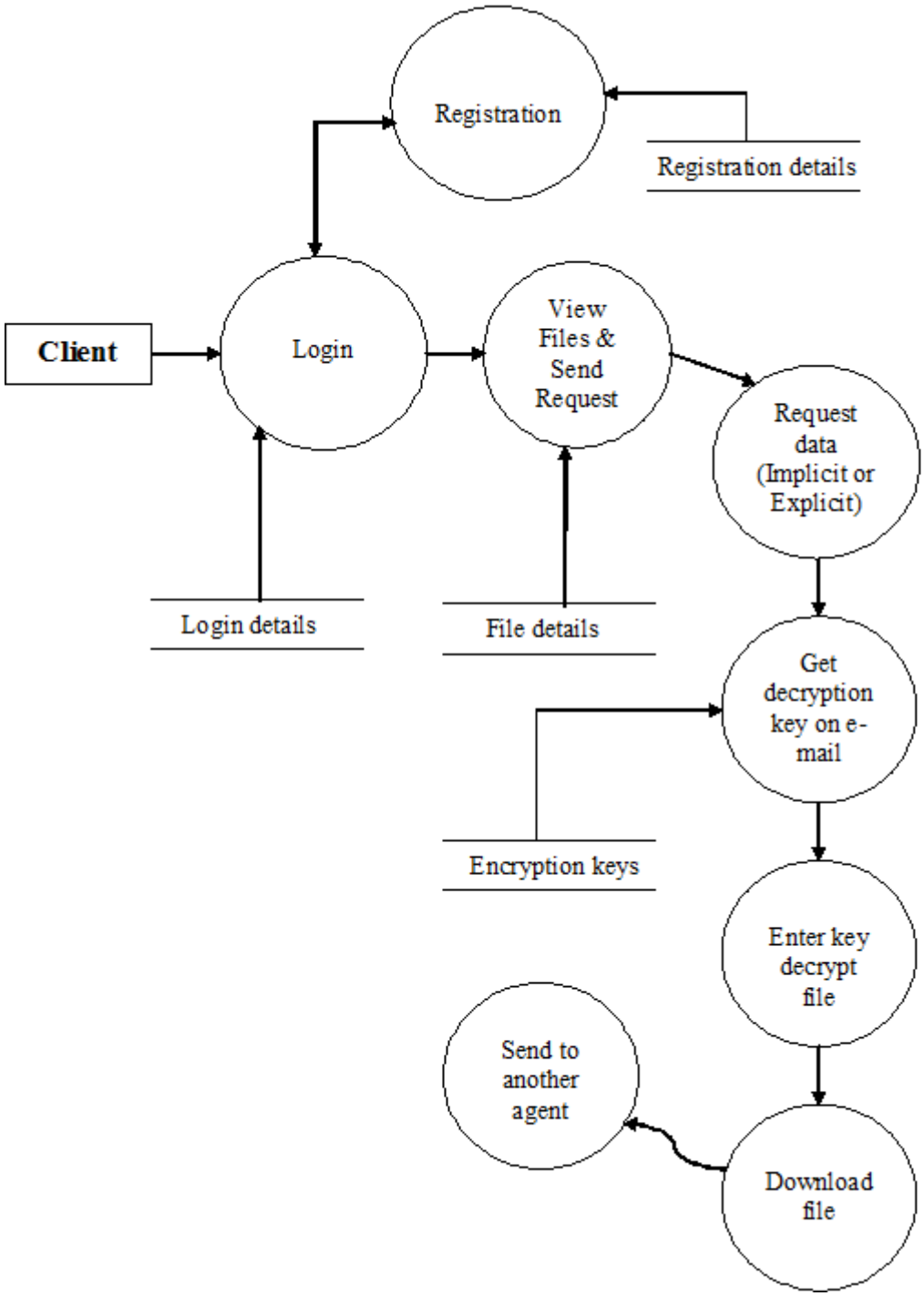


Figure 4: DataFlow2

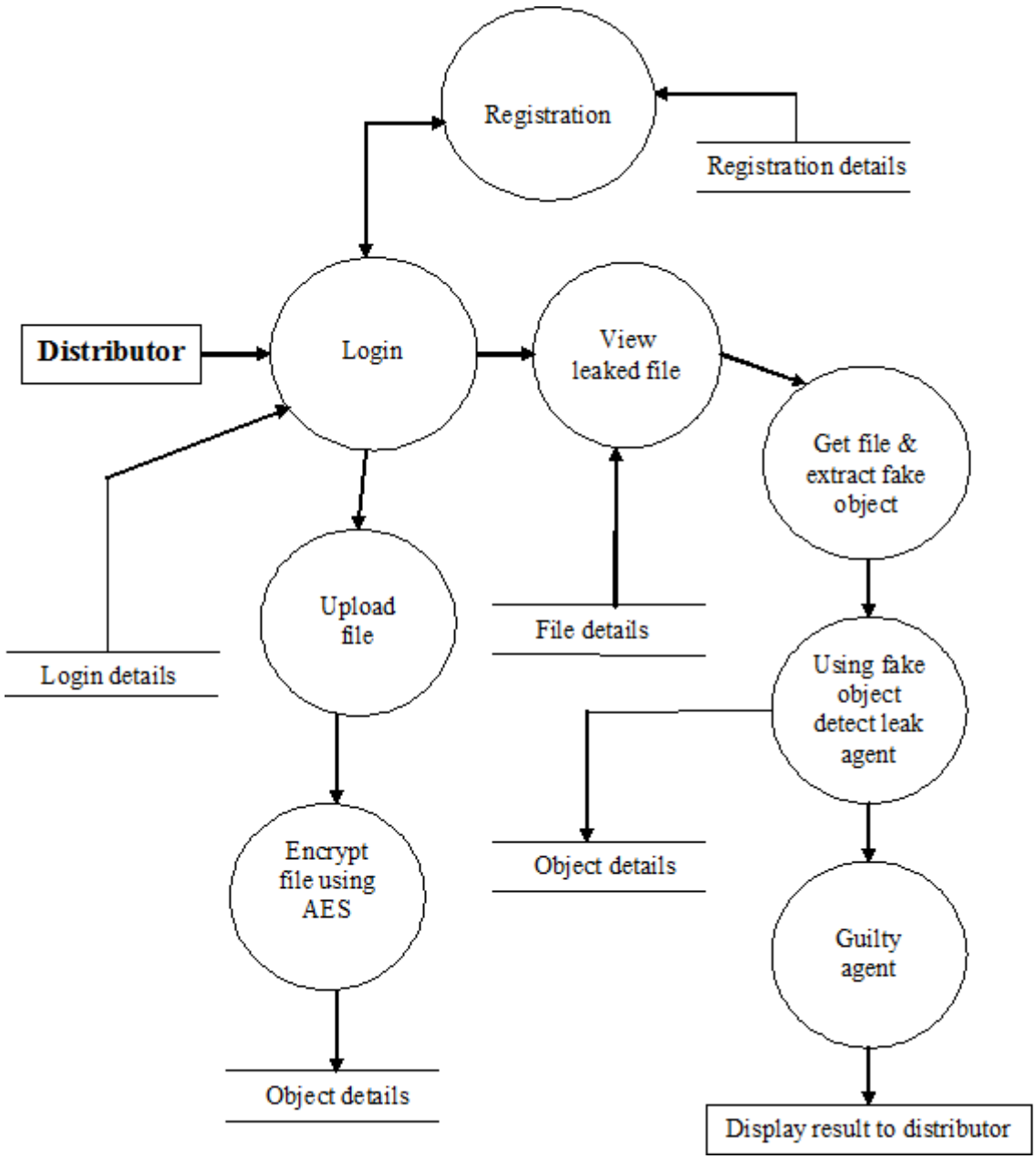


Figure 5: DataFlow3

6 Software Test Document

6.1 Introduction

6.1.1 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.

6.2 Test Plan Implementation

6.2.1 Black Box Testing

- Authorization
- Similarity of Documents
- Guilt Agent Identification

6.2.2 White Box Testing

For Cosine Similarity, Encryption and Decryption, Authentication.

- Calculation of Independent paths
- Loop Testing
- Logic Coverage

6.3 Test Case Design

Test Case Id	Module Name	Description	Input	Expected Output	Actual Output	Status(Pass/fail)
1	Registration	User Should be able to register	Username, Gender Email-Id, Password, Address	True: Successfully register False: Fail to register	True: Successfully register	Pass
2	Validation	Check if all the input are correct	Username, Gender Email-Id, Password, Address	True: Successfully register False: Invalid Input	True: Successfully register	Pass
3	Login	User Should be able to login in there account	Username, password	True: Successfully Login False: Invalid Credential	True: Successfully Login	Pass
4	Encryption Decryption	Should be able to encrypt and decrypt file	File	Successful: Generation of .aes file and decryption of main file Failure: Error	True: Successful Generation of .aes file and decryption of main file	Pass
5	Guilt Agent Report	Consist the name of Guilt Agent	File	Success: Guilt Agent Found. Failure: No Agent found Guilty	True: Guilt Agent Found	Pass

Figure 6: Test Case Design

7 Conclusion

Thus we have implemented the Initial configuration of Environment, Basic Coding and Inter-operability factors of the modules.We have verified the successful working of the encryption decryption and cosine similarity codes as well.Our next deliverable is to integrate these modules and deploy a test for the integrated system.