# A Project Report

on

## DETECTING DATA LEAKS VIA SQL INJECTION

***Submitted in partial fulfillment of the***
***requirement for the award of the degree of***

# Bachelor of Technology in Computer Science and Engineering


**Under The Supervision of**
**Mr. V. Arul**
**Assistant Professor**
**Department of Computer Science and Engineering**

**Submitted By**

18SCSE1010467-Kumari Suchitra

18SCSE1010639-Brajesh Kumar

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**GALGOTIAS UNIVERSITY, GREATER NOIDA,INDIA**

**DECEMBER- 2021**

# SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
# GALGOTIAS UNIVERSITY, GREATER NOIDA

## CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **"***DETECTING DATA LEAKS VIA SQL INJECTION***"** in partial fulfillment of the requirements for the award of the Bachelor of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the Sep-2021-March-2022, and Under supervision of **Mr V.Arul** Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida.

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

(Kumari Suchitra)

(Brajesh Kumar)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

(Mr V.Arul)
School of Computing Science and Engineering
Galgotias University
Greater Noida

## CERTIFICATE

The Final Project Viva-Voce examination of **18SCSE1010467- Kumari Suchitra, 18SCSE1010639-Brajesh Kumar** has been held on _____ and his/her work is Recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING.**

**Signature of Examiner(s)** **Signature of Supervisor(s)**

**Signature of Project Coordinator** **Signature of Dean**

Date:
Place: Greater Noida

# Acknowledgement

I express my deepest thanks to MR. V.ARUL, for giving necessary advices and guidance and arranged all facilities to make life easier. I choose this moment to acknowledge his contribution gratefully.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives.

Kumari Suchitra

Brajesh Kumar

# ABSTRACT

In the virtual and widely distributed network, the process of handover sensitive data from the distributor to the trusted third parties always occurs regularly in this modern world. It needs to safeguard the security and durability of service based on the demand of users A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). Some of the data are leaked and found in an unauthorized place (e.g., on the web or somebody's laptop). The distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. We propose data allocation strategies (across the agents) that improve the probability of identifying leakages. These methods do not rely on alterations of the released data (e.g., watermarks). In some cases, we can also inject "realistic but fake" data records to further improve our chances of detecting leakage and identifying the guilty party. The idea of modifying the data itself to detect the leakage is not a new approach. Generally, the sensitive data are leaked by the agents, and the specific agent is responsible for the leaked data should always be detected at an early stage. Thus, the detection of data from the distributor to agents is mandatory. This project presents a data leakage detection system using various allocation strategies and which assess the likelihood that the leaked data came from one or more agents For secure transactions, allowing only authorized users to access sensitive data through access control policies shall prevent data leakage by sharing information only with trusted parties and also the data should be detected from leaking by means of adding fake record`s in the data set and which improves probability of identifying leakages in the system. Then, finally it is decided to implement this mechanism on cloud server.

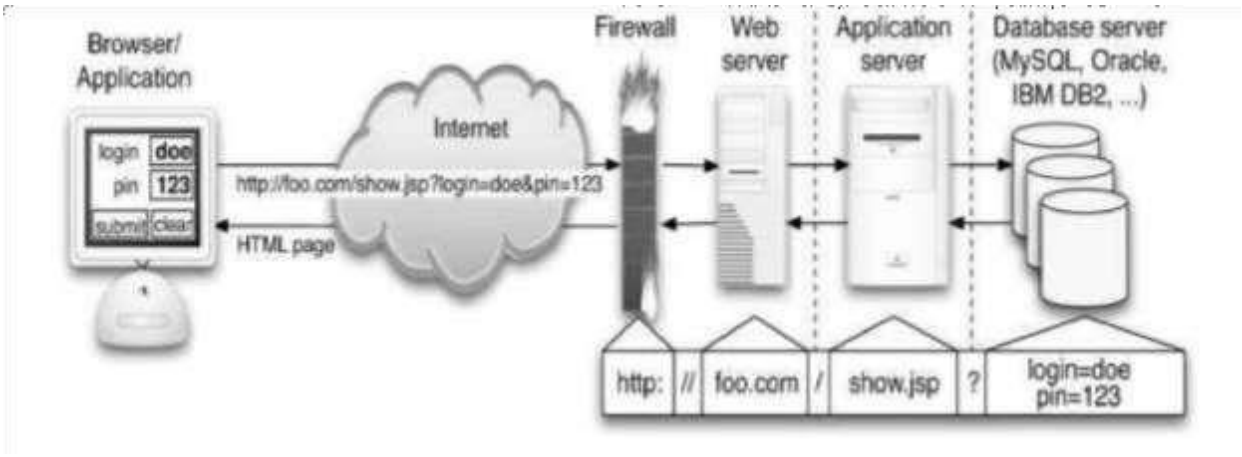
## CONTENT

## List of Figures

# CHAPTER-1

## Introduction

Web applications build SQL queries to access these databases based, in part, on user provided input. The intent is that Web applications will limit the kinds of queries that can be generated to a safe subset of all possible queries, regardless of what input users provide. However, inadequate input validation can enable attackers to gain complete access to such databases. One way in which this happens is that attackers can submit input strings that contain specially encoded database commands. When the Web application builds a query by using these strings and submits the query to its underlying database, the attacker's embedded commands are executed by the database and the attack succeeds. The results of these attacks are often disastrous and can range from leaking of sensitive data (for example, customer data) to the destruction of database contents. In this paper, we propose a new highly automated approach for dynamic detection and prevention of SQLI As. Intuitively, our approach works by identifying "trusted" strings in an application and allowing only these trusted strings to be used to create the semantically relevant parts of a SQL query such as keywords or operators.




Application access diagram

What is SQL injection?


SQL Injection

SQL is a **programming language designed to manage large amounts of data** stored in a database. It's primarily used to access, add, modify, and delete data from these databases. When a part of a website or application allows a user to input information turned directly into a SQL query, this makes the website vulnerable to SQL injection. **SQL injection is when malicious code is inserted as user input, so once it gets into the system and is turned into a SQL query, it begins to execute the malicious code**. Usually the purpose of this code is to access data to steal it (like user credentials) or delete it (to harm a business). If an attacker manages to access data and impersonate a database administrator, they can then access the entire system using those copied credentials.

## Aim of the Project

The aim of the project is to overcome data allocation problem and to send secured data for third party agent. Our goal is to detect when the distributor's sensitive data has been leaked by agents, and if possible to identify the agent that leaked the data. We develop unobtrusive techniquesfor detecting leakage of a set of objects or records.

## Objective

1. To detect when the distributor's sensitive data has been leaked by agents, and if possible, to identify the agent that leaked the data.
2. To secure our data from a third party.

## Scope of Project

Perturbation is a very useful technique where the data is modified and made “less sensitive” before handed of agents. We develop unobtrusive techniques for detecting leakage a set objects or records.

We develop a model for assessing the “guilt” od agent. We also present algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker. Finally, we also consider the option of adding “fake” objects to the distributed set. Such objects do not correspond to real entities but appear realistic to the agent. In a sense, the fake object acts as a type of watermark for the entire set, without modifying any individual member. If it turns out an agent was given one or more fake object that were leaked, then the distributor can be more confident that agent was guilty.

## 1.2 PROBLEM IDENTIFICATION

SQL injection errors occur when:

1. Data enters from an untrusted source.
2. Dynamically construct a SQL query.

The main consequences are:

- **Confidentiality:** An application databases will hold sensitive data, loss of confidentiality is a most frequent problem with SQL Injection .
- **Authentication:** If poor SQL commands are used to check user names and passwords, it may be possible to connect to a system as another user with no previous knowledge of the password.
- **Authorization**: If authorization information is held in a SQL database, it may be possible to change this information through the successful exploitation of a SQL Injection vulnerability.
- **Integrity:** Just as it may be possible to read sensitive information, it is also possible to make changes or even delete this information with a SQL Injection attack

### 1.2.1 Problem description:

DLD is the system such that DLD={A,,D,T,U,R,S,U*,C,M,F}.

.

- {A} is the Administrator who will be given all the privileges about the application activities such as blocking the guilty agent using probability distribution table. He can also see the shared files and manages the whole database
- {D} is the Distributor who will send data T to different agents U .He will also accept requests from the different agents about the particular type of files and will distribute it accordingly
- T is the set of Data object that are supplied to agents. T can be of any type and size, e.g., they could be tuples in a relation, or relations in a database. Specifically in our project we are sending the text files.

  $T= \{t_1,t_2,t_3,...t_n\}$

- U is the set of Agents who will receive the data from the distributor D. The agent can also request for the particular type of text files. These agents are continuously monitored by the administrator and distributor in case they leak the data. $U=\{u_1,u_2,u_3,...u_n\}$
- R is the Record set of data objects which is sent to agents. This data is sent to agent depending upon the distributor's choice or agent's request.

  $R=\{t_1,t_3,t_5..tm\}$ **R is a subset of T**

- S is the Record set of data objects which are leaked by the particular agent $U_i$ to third party. This agent who has leaked the data will be further referred as a guilty agent and will be blocked by the admin.

  $S=\{t_1,t_3,t_5..tm\}$ **S is a Subset of T**

- U* is the set of all agents which may have leaked the data. They are referred as Guilty Agents

  $U^*=\{u_1,u_3,...u_m\}$ **U* is a subset of U**

- C is the set of Conditions which will be given by the agents to the distributor

$C=\{cond_1,cond_2,cond_3,...,cond_n\}$

- M is set of data objects to be send in Sample Data Request algorithm
  $M=\{m_1,m_2,m_3,...,m_n\}$

- F is the set of fake o.

## Tool and Technology Used

**Software requirements:**

- Operating system : Windows 7 / 8 / 8.1 / 10
- Coding Language : My SQL, PHP ,Java script, bootstrap5

  Database : MySQL v5.1

- 
- Application Server :XAMPP Server

## 1.3 PROBLEM DEFINITION

Our goal is to detect when the distributor's sensitive data have been leaked by agents, and if possible to identify the agent that leaked the data.

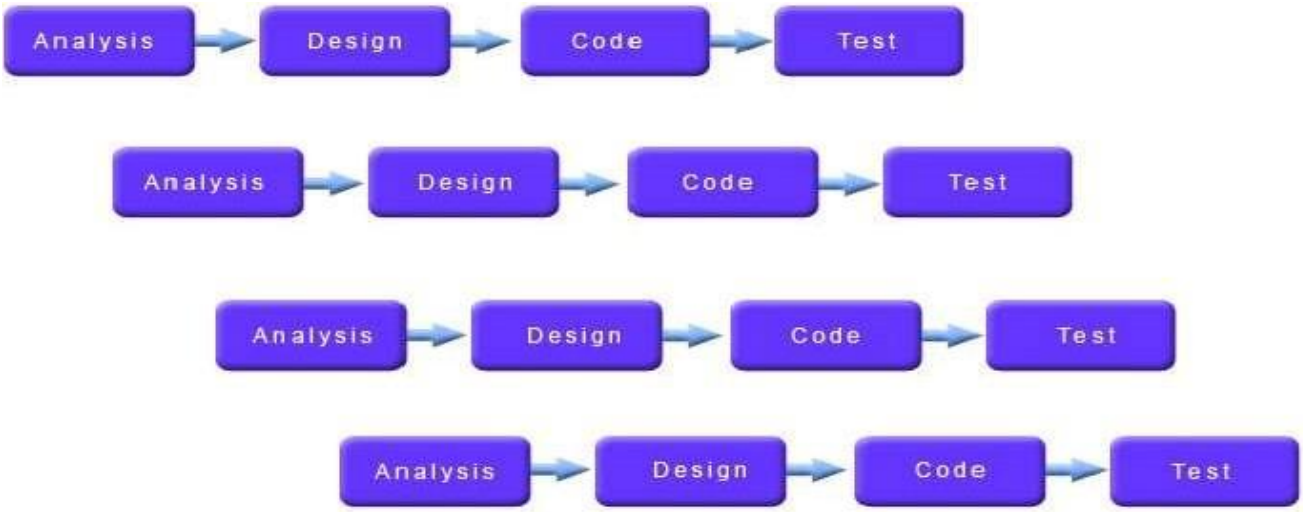
## 1.4 APPLYING SOFTWARE ENGINEERING APPROACH

Incremental model is an evolution of waterfall model. The product is designed, implemented, integrated and tested as a series of incremental builds. It is a popular model software evolution used many commercial software companies and system vendor.

Incremental software development model may be applicable to projects where: Software

Requirements are well defined, but realization may be delayed. The basic software

functionality are required early




**Advantages**

- Generates working software quickly and early during the software life cycle.
- More flexible - less costly to change scope and requirements.
- Easier to test and debug during a smaller iteration.
- Easier to manage risk because risky pieces are identified and handled during its iteration.

**Disadvantages**

- Each phase of an iteration is rigid and do not overlap each other.
- Problems may arise pertaining to system architecture because not all requirements are gathered upfront for the entire software life cycle.

CHAPTER-2

# Literature Survey

**Sandip A. Kale** describes the results of implementation of Data Leakage Detection Model. Currently watermarking technology is being used for the data protection. But this technology doesn't provide the complete security against date leak age. This paper includes the difference between the watermarking & data leakage detection model's technology. This paper leads for the new technique of research for secured data transmission & detection, if it gets leaked.

**S Ramkumar et al**, investigate and utilized the characteristic of the group movement of objects to explore the group relationship and tracking them. The goal is to efficiently mine the group movement activity using clustering and sequential pattern mining. Clustering was applied to find both groups of similar teams and similar individual members. Sequential pattern mining was used to extract sequences of frequent events. To enable the continuous monitoring the group object movement, the system introduces a special technique called minor clustering and Cluster Assembling algorithm. Several solutions on route were implemented, but those methods were energy consumed. In order to reduce the energy, the proposed system used data mining methods to effectively handle the group movement of objects.

**Chandni Bhatt et al**," study a data distributor has given sensitive data to a set of supposedly trusted agents. Sometimes data is leaked and found in unauthorized place e.g., on the web or on somebody's laptop. For example, a hospital may give patient records to researchers who will devise new treatments. Similarly, a company may have partnerships with other companies that require sharing customer data. Another enterprise may outsource its data processing, so data might be given to various other companies. The owner of the data is called as distributors and the trusted third parties are called as agents. Data leakage happens every day when confidential business information such as customer or patient data, company secrets, budget information etc. is leaked out. When this information is leaked out, then the companies are at serious risk. Most probably data are being leaked from agent's side. So, company has to very careful while distributing such a

data to agents. The Goal of Our project is to analyse “how the distributer can allocate the confidential data to the Agents so that the leakage of data would be minimized to a Greater Extent by finding a guilty agent”.

## Garcia-Molina [2010]

Our scenario is based on the scenario presented by Papadimitriou and Garcia-Molina [2010], with several modifications. In their paper, Papadimitriou and Garcia presented a method for data leakage detection. In the scenario that they address, a distributor distributes sensitive data to almost all agents according to a specific request that is issued for each one of the agents. An example of such a scenario is a proactive CRM system in which the data owner decides which customer or stakeholder to call, and the customer or stakeholder details are forwarded to the third party call agent. If sensitive data is leaked, the data owner would like to be able to identify the source of leakage, or at least be able to estimate the likelihood of each agent to have been involved in the incident. Therefore, A guilt model is proposed for estimating the probability that an agent is involved in a given data leakage. The capability and ability to identify the of the leakage depends on the distribution of data objects among the agents. Therefore, a data allocation method that distributes data records among the agents based on the agents’ requests and optimization models are presented. The proposed allocation method ensures that object sharing among the agents is minimal, and therefore, in the case of a leakage incident, the data owner will be able to use the guilt model to identify the source of leakage with high probability.

## Amol O. Gharpande et al

gives review idea about data leakage detection techniques. A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). Some of the data is leaked and found in an unauthorized place (e.g., on the web or somebody’s laptop). The distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. We propose data allocation strategies (across the agents) that improve the probability of identifying leakages. These methods do not rely on alterations of the released data (e.g., watermarks). In some case, we can also inject “realistic but fake” data records to further improve our chances of detecting leakage and identifying the guilty party.

## Garcia-Molina [2010]

Two types of data requests are being considered in Papadimitriou and Garcia-Molina [2010]: explicit request and sample request. An explicit request contains predefined conditions, and all the objects in the dataset that comply with these conditions must be returned. A sample request defines the amount of objects to be randomly selected from the entire dataset. Combined requests (i.e., requests for a sample of objects that comply with a predefined condition) are not handled by the proposed algorithms; However, it is explained how they might be handled using the proposed algorithms. They as well proposed including fake data to the lists of real data objects when distributing them to the agents. Fake data objects may help to better distinguish between the agents and increase the accuracy of the guilt model (e.g., when each untrusted agent receives a unique fake object). Four scenarios can be stated by the two request types (sample or explicit) and the two options of planting fake objects in the result sets (using or not using fake objects). It is assumed that in each scenario, all of the agents issue the same type of requests (i.e., either explicit or sample queries), and if fake objects are up to use, the same amount of objects will be planted for all agents. Several allocation algorithms are proposed to deal with each scenario. Empirical evaluation showed that the proposed algorithms reached a significantly greater ability to identify the source of leakage (compared with simple allocation algorithms), even in cases where there was a large overlap between the objects that the agents receiv

# Project Design

2.2.1. PROPOSED SYSTEM

The architecture and flow of the proposed system in Figure

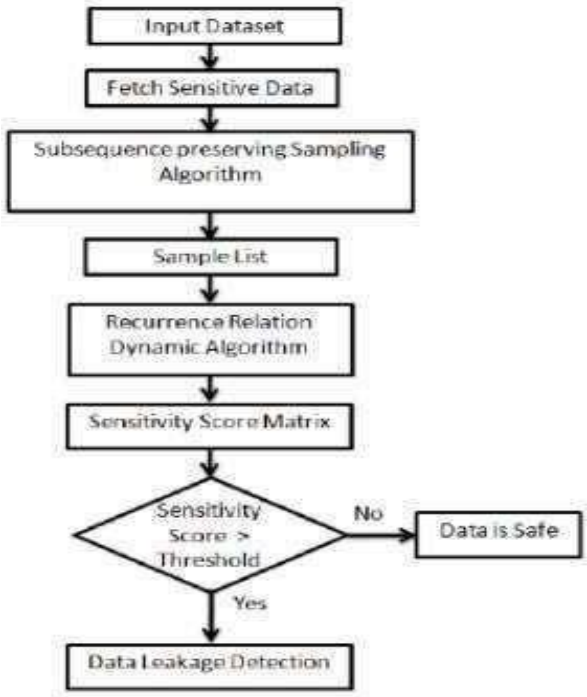



Fig3:The architecture and flow of the proposed system

The proposed system initially user browses the input dataset and fetches the sensitive data from input dataset. For generating the sample list, subsequence preserving samplingalgorithm is used to generate sampling algorithm.

If given string is p and its substring is q, this is denoted by p <=q), then p 0 is also a substring of q 0 (p 0 <=q 0), where p 0 is a sensitive data of p sample and sampled sensitive data sample of p, and q 0 is a sensitive sampled data of q. After that matrix forsensitivity score is calculated by recurrence relation dynamic programming algorithm.After that system is compared threshold value with sensitivity score value if sensitivity score is greater than threshold value then data leak is detected, otherwise data is not leak.

- Our goal is to detect when the distributor's sensitive data has been leaked by agents, and if possible to identify the agent that leaked the data. Perturbation is a very useful technique where the data is modified and made "less sensitive" before being handed to agents. we develop *unobtrusive* techniques for detectingleakage of a set of objects or records.
- In this section we develop a model for assessing the "guilt" of agents. We also present algorithms fordistributing objects to agents, in a way that improves our chances of identifying a leaker.
- Finally, we also consider the option of adding "fake" objects to the distributed set. Such objects do notcorrespond to real entities but appear realistic to the agents.
- In a sense, the fake objects acts as a type of watermark for the entire set, without modifying any individual members. If it turns out an agent was given one or more fake objects that were leaked, then thedistributor can be more confident that agent was guilty.

## Advantage Proposed System:-

- We can provide security to our data during its distribution or transmission and even wecan detect if that gets leaked
- we have presented implement a variety of data distribution strategies that can improvethe distributor's chances of identifying a leaker
- Quick response time
- Customized processing
- Small memory factor
- Highly secure
- Replication in Heterogenic Database
- Easy updating.

## FAKE OBJECTS

The distributor may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. However, fake objects may impact the correctness of what agents do, so they may not always be allowable. The idea of perturbing data to detect leakage is not new. However, in most cases, individual objects are perturbed, e.g., by adding random noise to sensitive salaries, or adding a watermark to an image. In our case, we are perturbing the set of distributor objects by adding fake elements. In some applications, fake objects may cause fewer problems that perturbing real objects.

For example, say that the distributed data objects are medical records and the agents are hospitals. In this case, even small modifications to the records of actual patients may be undesirable. However, the addition of some fake medical records may be acceptable, since no patient matches these records, and hence, no one will ever be treated based on fake records. Our use of fake objects is inspired by the use of "trace" records in mailing lists. In this case, company A sells to company B a mailing list to be used once (e.g., to send advertisements). Company A adds trace records that contain addresses owned by company A. Thus, each time company B uses the purchased mailing list, A receives copies of the mailing. These records are a type of fake Objects that help identify improper use of data.

The distributor creates and adds fake objects to the data that he distributes to agents. We let Fi _ Ri be the subset of fake objects that agent Ui receives. As discussed below, fake objects must be created carefully so that agents cannot distinguish them from real objects.

## PRIVACY, PRESERVATION AND PERFORMANCE (THE 3 P'S)

There are several attributes to consider when designing a dis-tributed data management system. For example, we would like a system that enforces "privacy" by not divulging data to unau-thorized entities. At the same time, we want to "preserve" the data effectively i.e., protect it from hardware failures, natural disasters, and so on. And, of course, we do not want to sacri-fice "performance" – a system that runs slowly may not be very useful. There are many such desirable attributes: confidential-ity, integrity, reliability, availability, throughput, and so on.

While each of these attributes has been studied extensively, there is not much work that considers the tradeoffs between them. We believe that today the real challenge in designing a system is in achieving a balance between several conflicting at-tributes. For example, a system that simply deletes all input data would be very "secure" – no data will ever leak out! – but would be unattractive in other dimensions. Similarly, one can build a highly "reliable" system that proliferates many copies of its data. This system would excel along the preservation dimension, but each extra copy would increase the chances of unauthorized break-ins. If we encrypt a large collection of records as a single "blob", performance for reading individual records suffers. If we encrypt each record individually, reads will be faster, but overall security may not be as strong.

In this paper we study, in a unified way, three conflicting attributes of a distributed system, and show how to design systems that strike a balance among them. We refer to these attributes as the "3 P's" of distributed data management – privacy, preservation and performance. Informally:

- Privacy refers to protecting data from unauthorized access (related to security and confidentiality).
- Preservation ensures that data is still available and uncorrupted far into the future (related to integrity,availability, and reliability).
- Performance refers to the quick, timely access to our distributed data (related to response time andthroughput).

**The main contributions are:**

A unified framework for measuring the privacy, preservation and performance offered by a system.

1. An algorithm for finding systems that achieve a desired balance between the 3 P's.

Our work represents a bridge between system design and risk management. Risk managementis the science of identifying, measuring and mitigating the uncertainty related to threats.

Privacy is measured by the damage, D, caused by leakage of in-formation to unauthorized parties,

We identify privacy violations, data loss and poor performance as threats faced by a system in its

daily operation. We then measure the potential harm caused by these threats:

a) Preservation is measured by the loss, L, incurred due to lost or corrupted data, and

b) Performance is measured by the running time, T, of read and writes commands issued to the system.

We mitigate risk not by providing "guarantees" – instead, we accept that bad things can happen, and try to minimize the harm when bad things do happen. How can we ensure that we do well "on average", and how can we reduce the likelihood of "catastrophes"? We treat D, L and T as random variables, and solve optimization problems involving their means and variances.

To effectively manage risk, it is crucial to account for the heterogeneity within a collection of data objects. For exam-ple, there may be a great deal of damage done if an internal e-mail discussing corporate strategy is leaked to outsiders. But, nobody may care if the e-mail was accidentally deleted. On the other hand, nobody would mind if marketing materials were "leaked" to outsiders (it's probably a good thing!). But if these marketing materials were lost, the time and money spent developing them would be lost. Moreover, email is mostly text whereas marketing flyers are filled with large images, so the performance implications are vastly different between the two. In our framework, we explicitly model data objects in a heterogeneous collection as having differing values, sensitivities, sizes and usage patterns.

The outcome of our work is a solution strategy for the following problem: Given a set of resources (e.g., data centers, servers, storage devices), and a collection of data objects (e.g., documents, photos, MP3s, e-mail) with known usage characteristics, sensitivities and sizes, how should we distribute them across our resources to achieve a desired balance between the 3 P's – privacy, preservation and performance?

# CHAPTER-3

# Working of Project

## 3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities.

The following are its features:

## TECHNICAL FEASIBILITY

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

Does the existing technology is sufficient for the suggested one? Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So, there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

## BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible

## ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposedsystem. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

## 3.2 TECHNOLOGY DETAILS USED IN PROJECT

### 3.2.1 MYSQL

MySQL is the world's most used relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. The SQL phrase stands for Structured Query Language. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python"

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools or download MySQL front-ends from various parties that have developed desktop software and web applications to manage MySQL databases, build database structures, and work with data records.

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

There are however limits to how far performance can scale on a single server, so on larger scales, multi-server MySQL deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations. The master server synchronizes continually with its slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in

performance can be achieved by caching the results from database queries in memory using memcached, or breaking down a database into smaller chunks calledshards which can be spread acrossa number of distributed server clusters.

### 3.2.2 XAMPP SERVER

XAAMP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perlprogramming languages.

Installing XAMPP takes less time than installing each of its components separately. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from onecomputer to another.

It is offered in both a full, standard version and a smaller version.

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. In practice, however, XAMPP is sometimes used to actually serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package XAMPP also provides support for creating and manipulating databases in MySQL and SQLite among others.

## 3.3 SYSTEM DESIGN AND DIAGRAM

### 3.3.1 DATA FLOW DIAGRAM

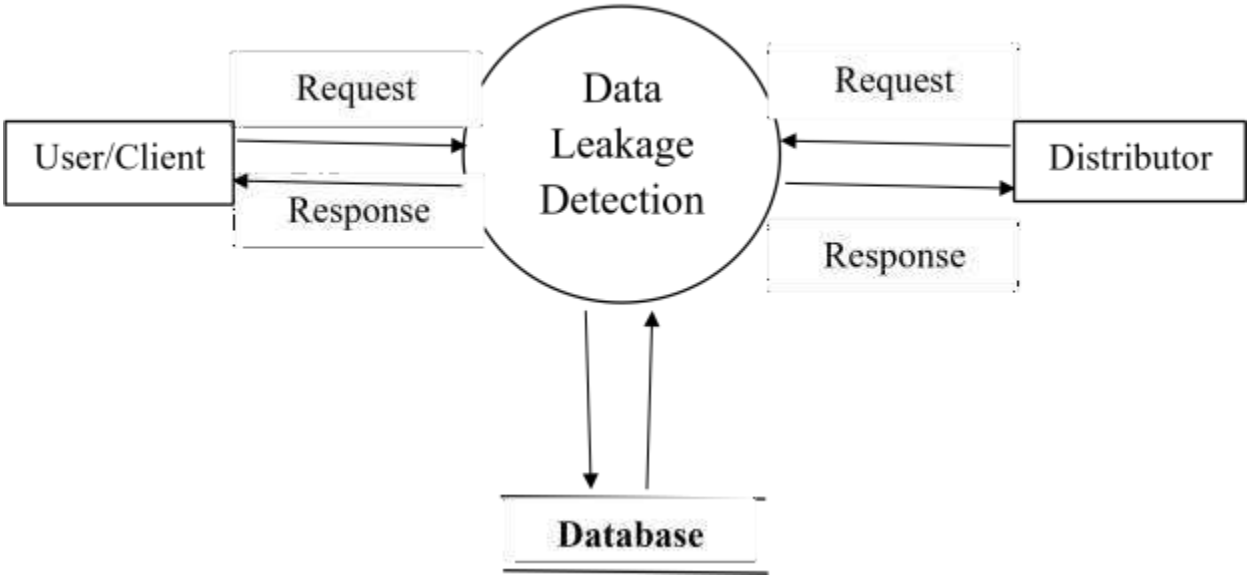
A data flow diagram (DFD) is a graphical representation of the flow of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then exploded to show more detail of the system being modelled.

Symbols:

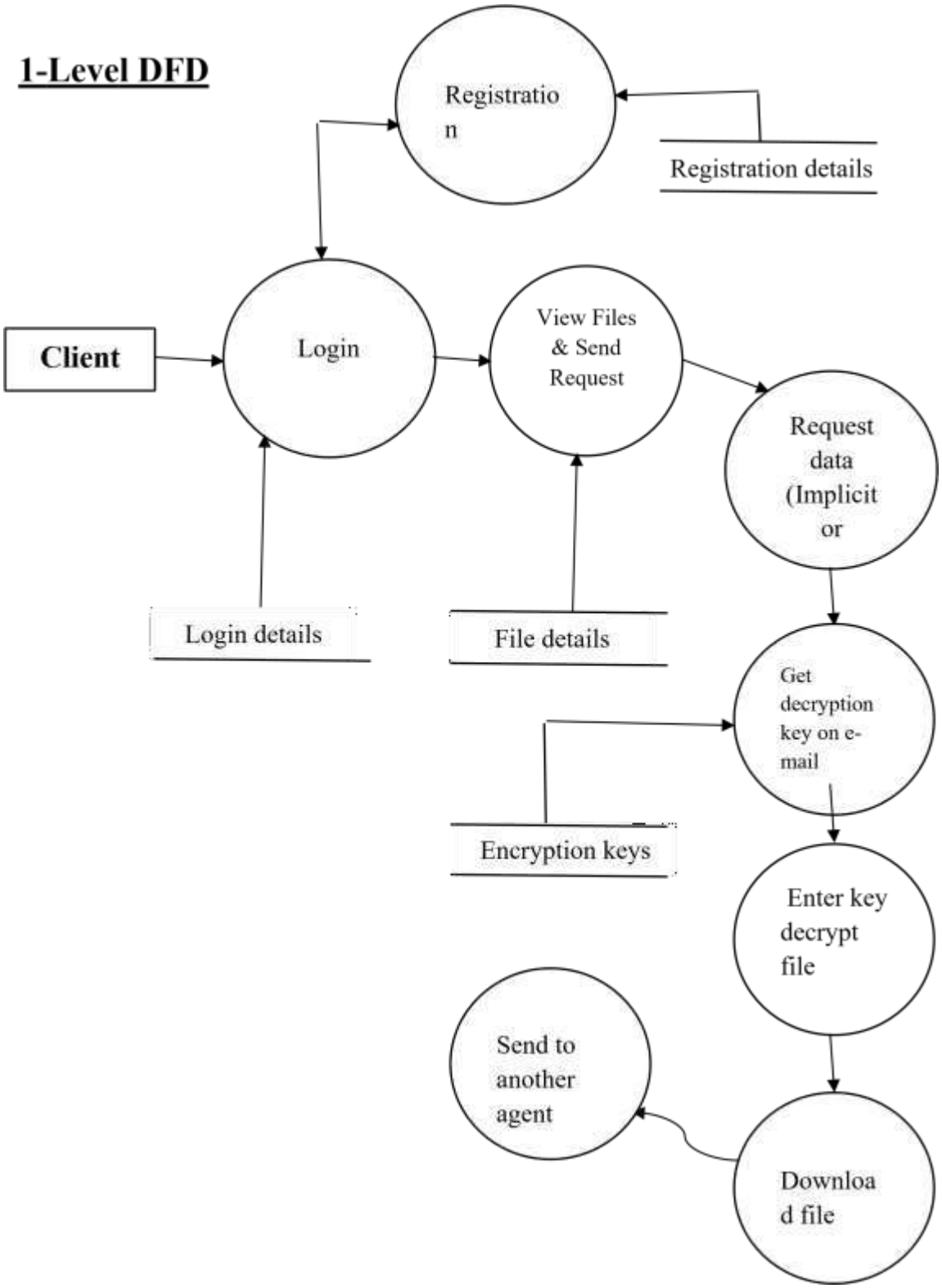
The four components of a data flow diagram (DFD) is:

- External Entities/Terminators are outside of the system being modelled. Terminators represent where information comes from and where it goes. In designing a system, we have no idea about what these terminators do or how they do it.
- Processes modify the inputs in the process of generating the outputs.
- Data Stores represent a place in the process where data comes to rest. A DFD does not say anything about the relative timing of the processes, so a data store might be a place to accumulate data over a year for the annual accounting process.

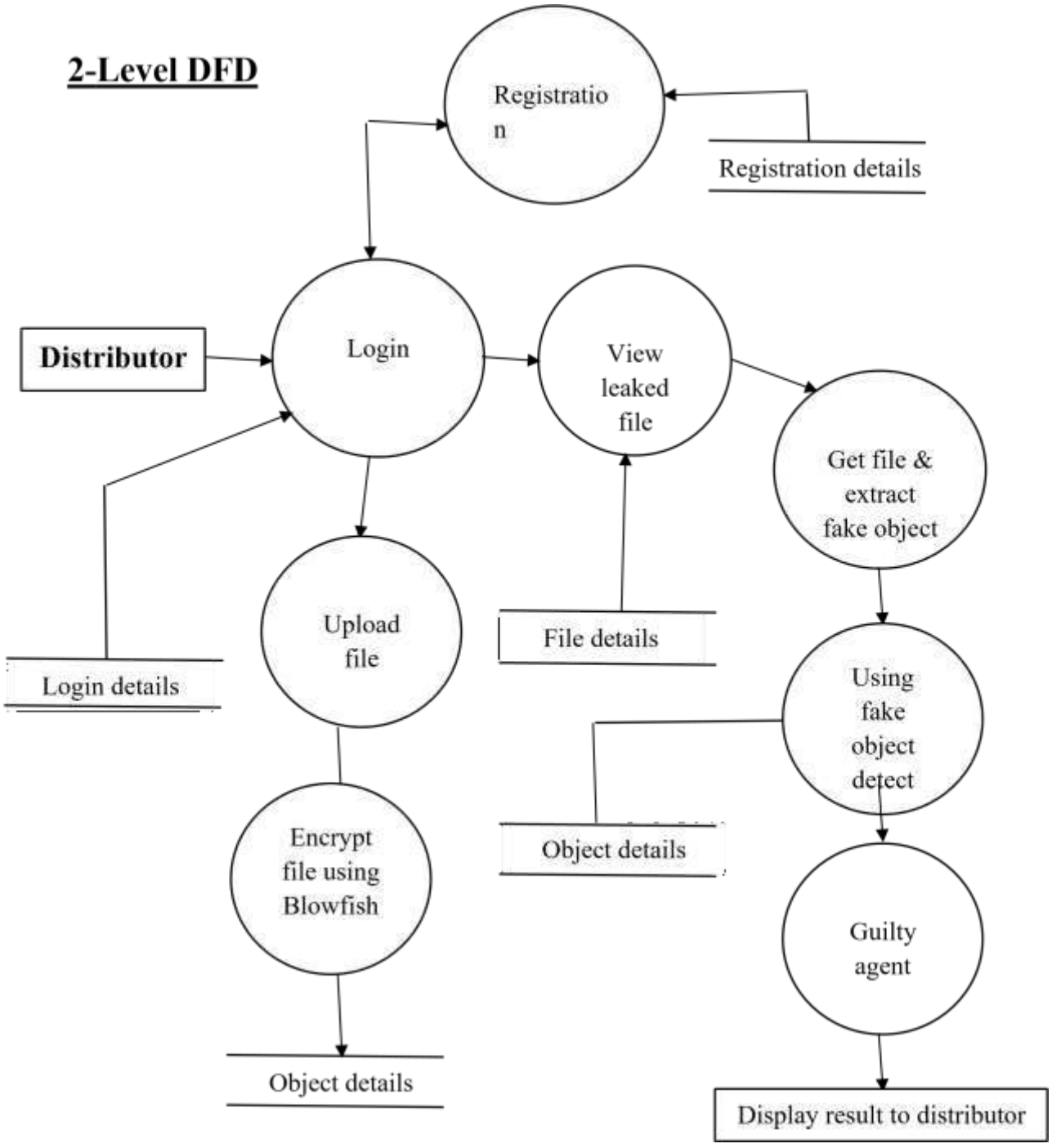
**0-LEVEL DFD**



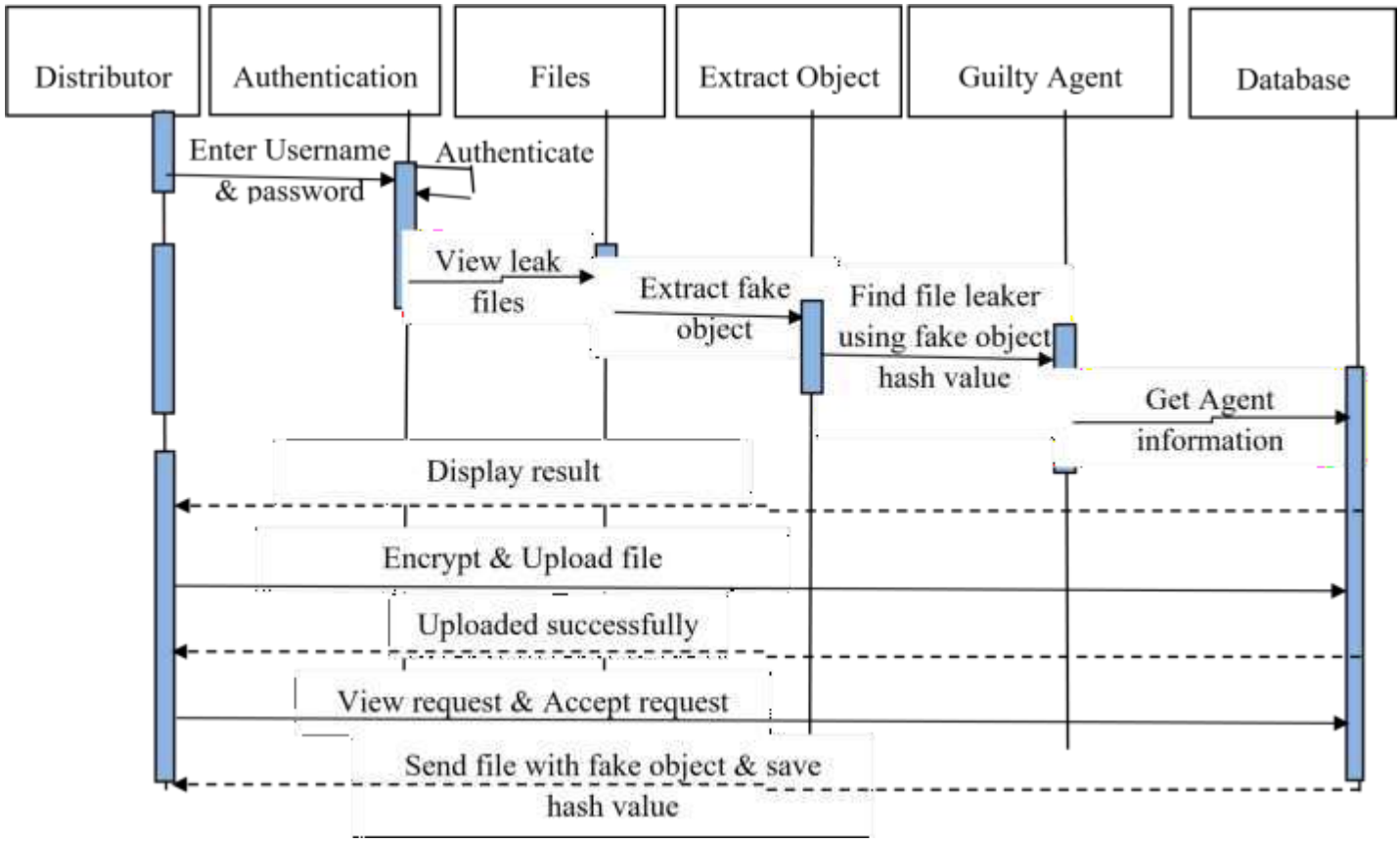
## 1-Level DFD


Registratio
n
Registration details
Client
Login
View Files
& Send
Request
Request
data
(Implicit
or
Login details
File details
Get
decryption
key on e-
mail
Encryption keys
Enter key
decrypt
file
Send to
another
agent
Downloa
d file

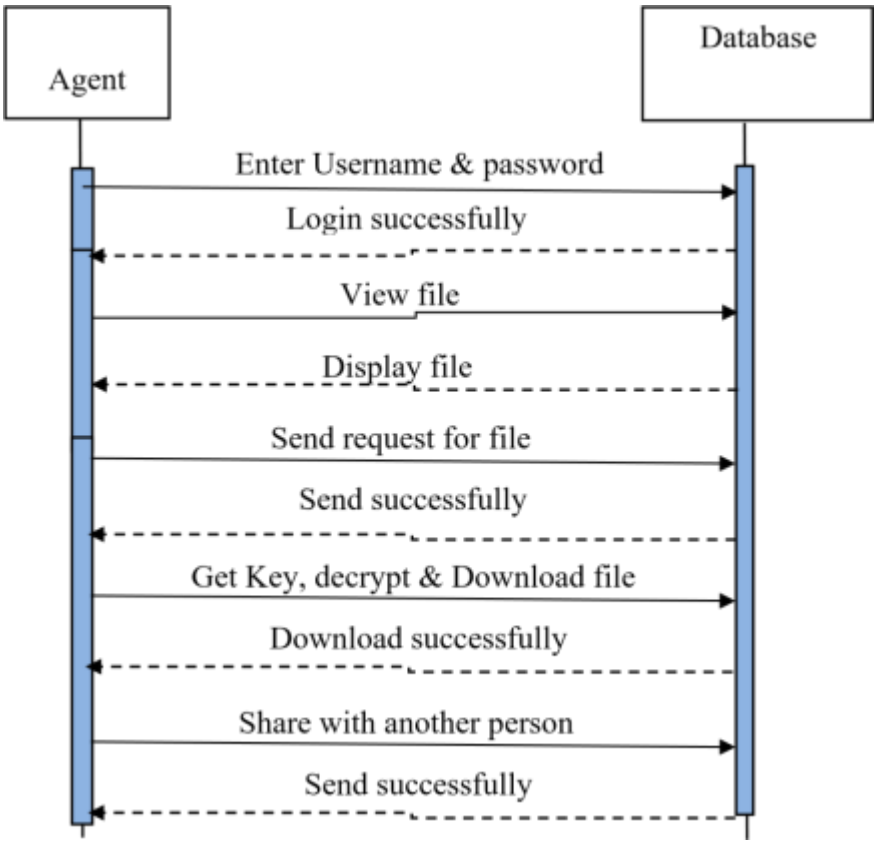
## 2-Level DFD


Registratio
n
Registration details
Distributor
Login
View
leaked
file
Get file &
extract
fake object
Upload
file
File details
Login details
Using
fake
object
detect
Encrypt
file using
Blowfish
Object details
Guilty
agent
Object details
Display result to distributor

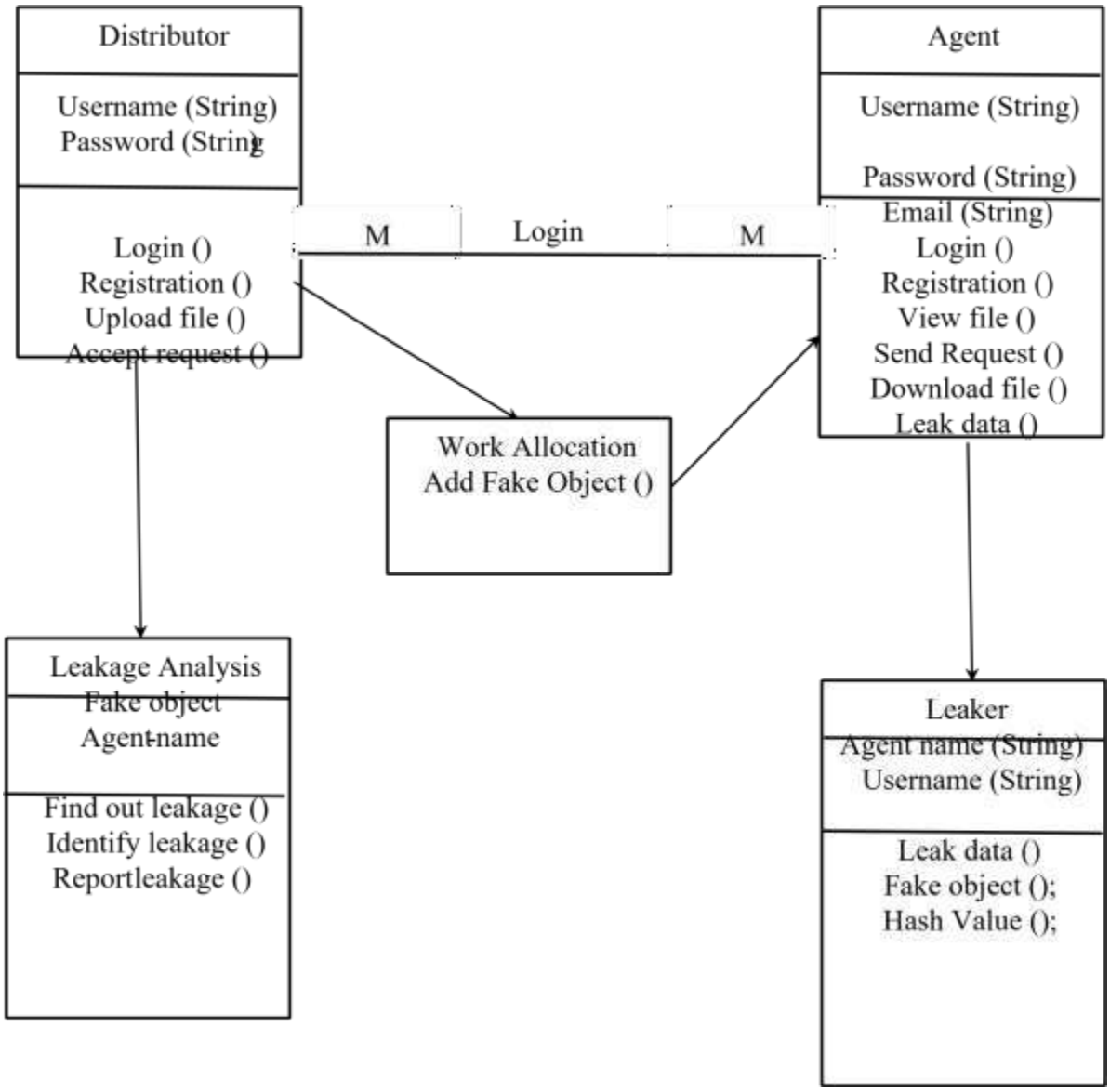
## Distributor Sequence Diagram



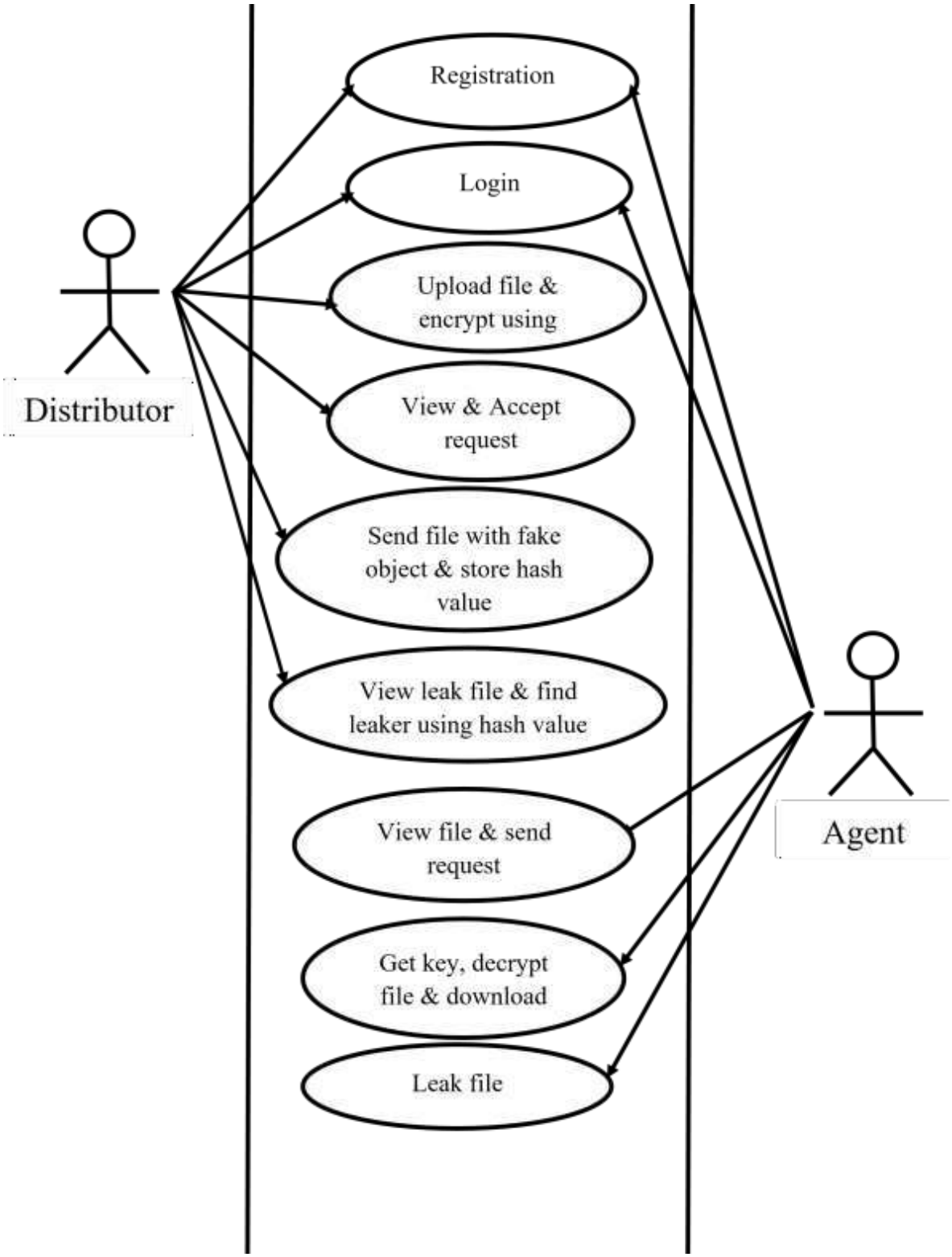
## Agent Sequence Diagram



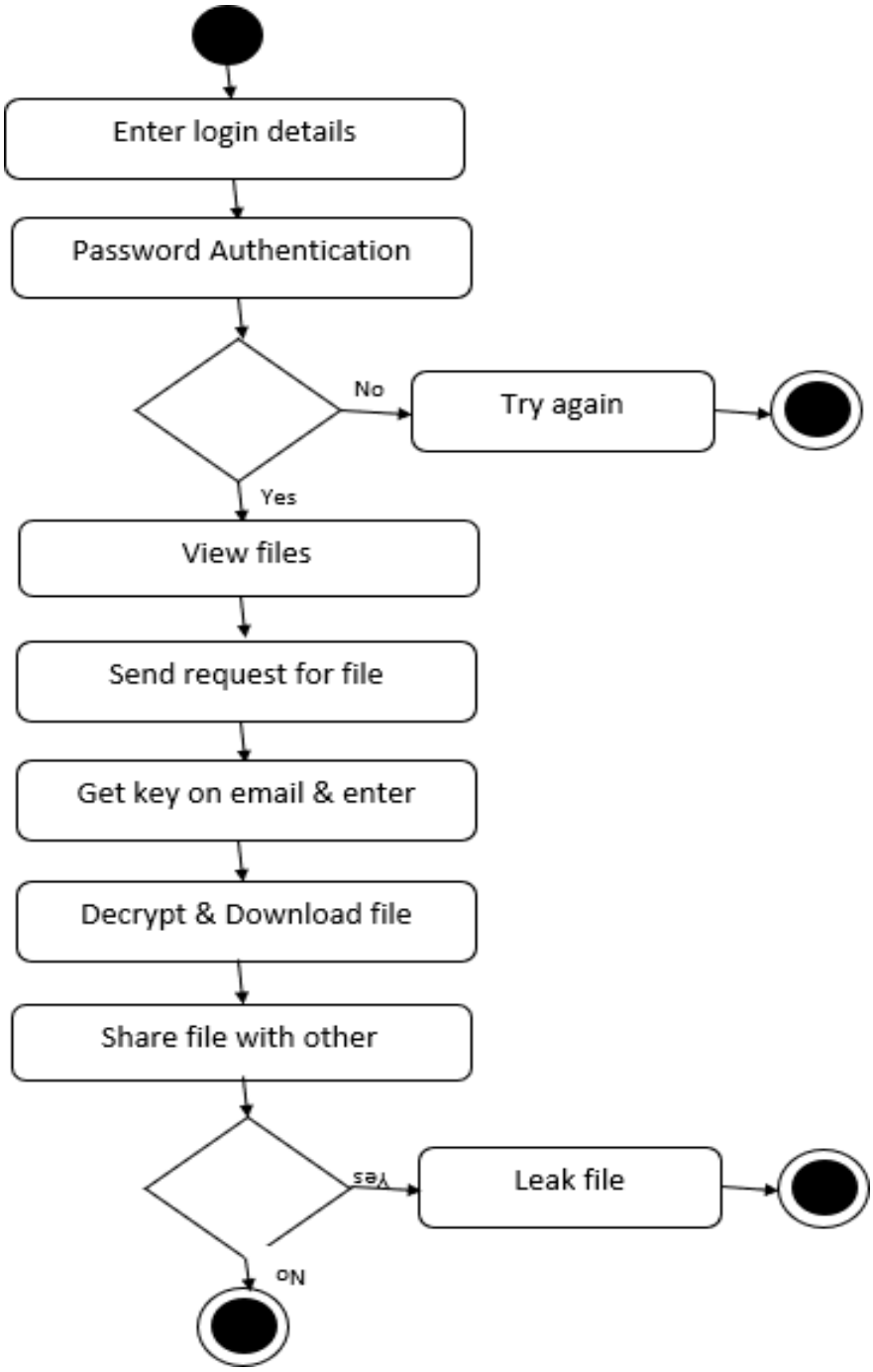
## Class Diagram



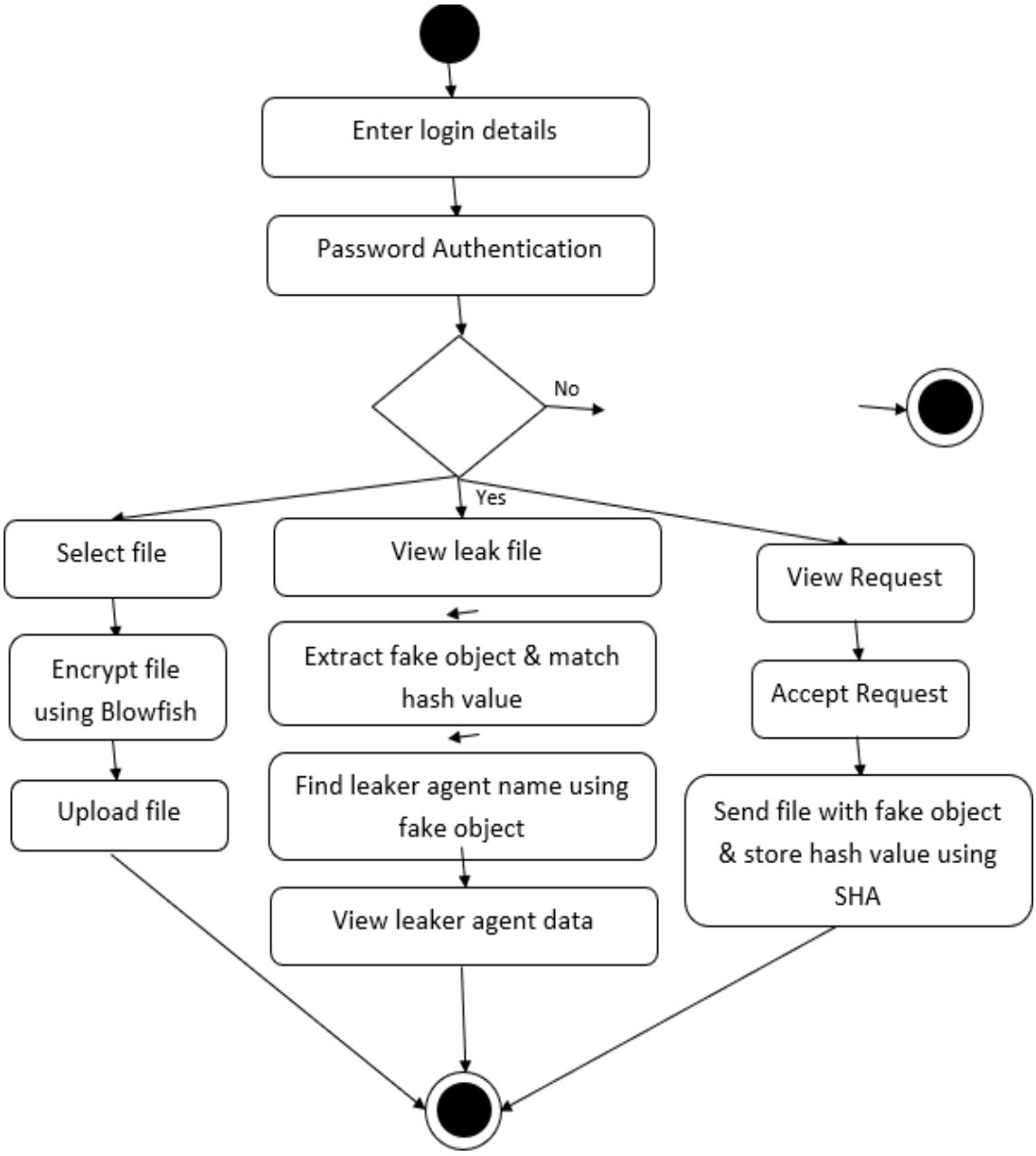
## Use Case Diagram



Activity diagram for User:



Activity diagram for Owner:

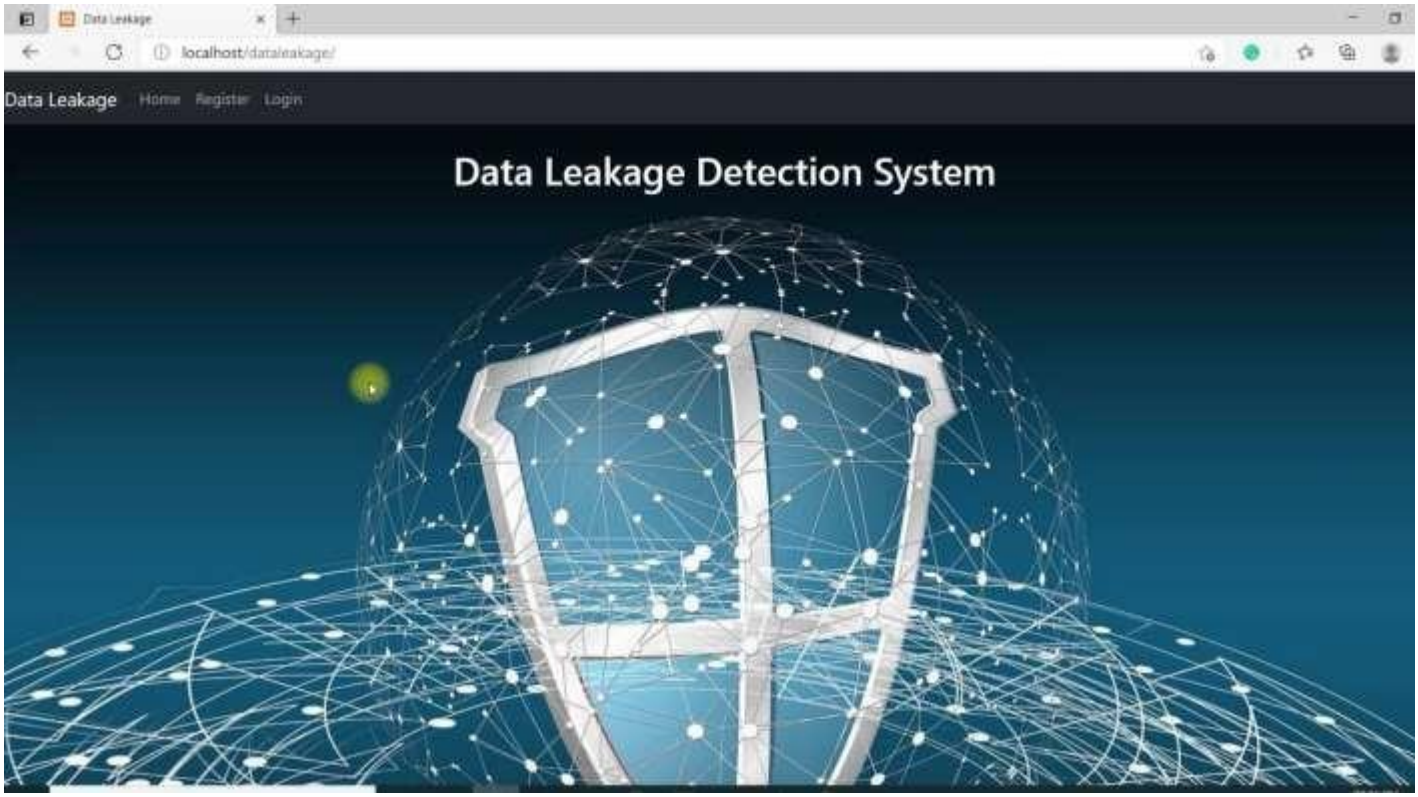


# CHAPTER-4

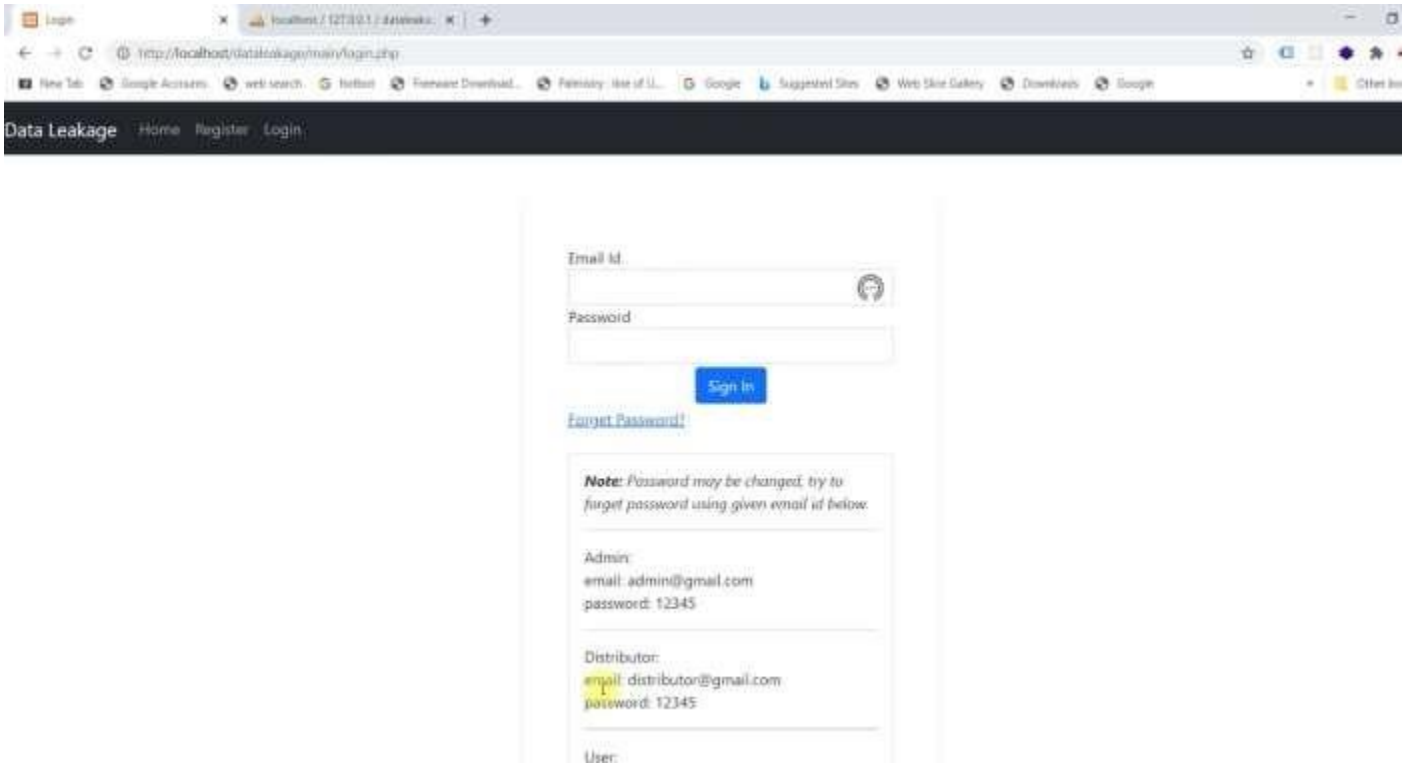
## Results and Discussion

### Screen Shot

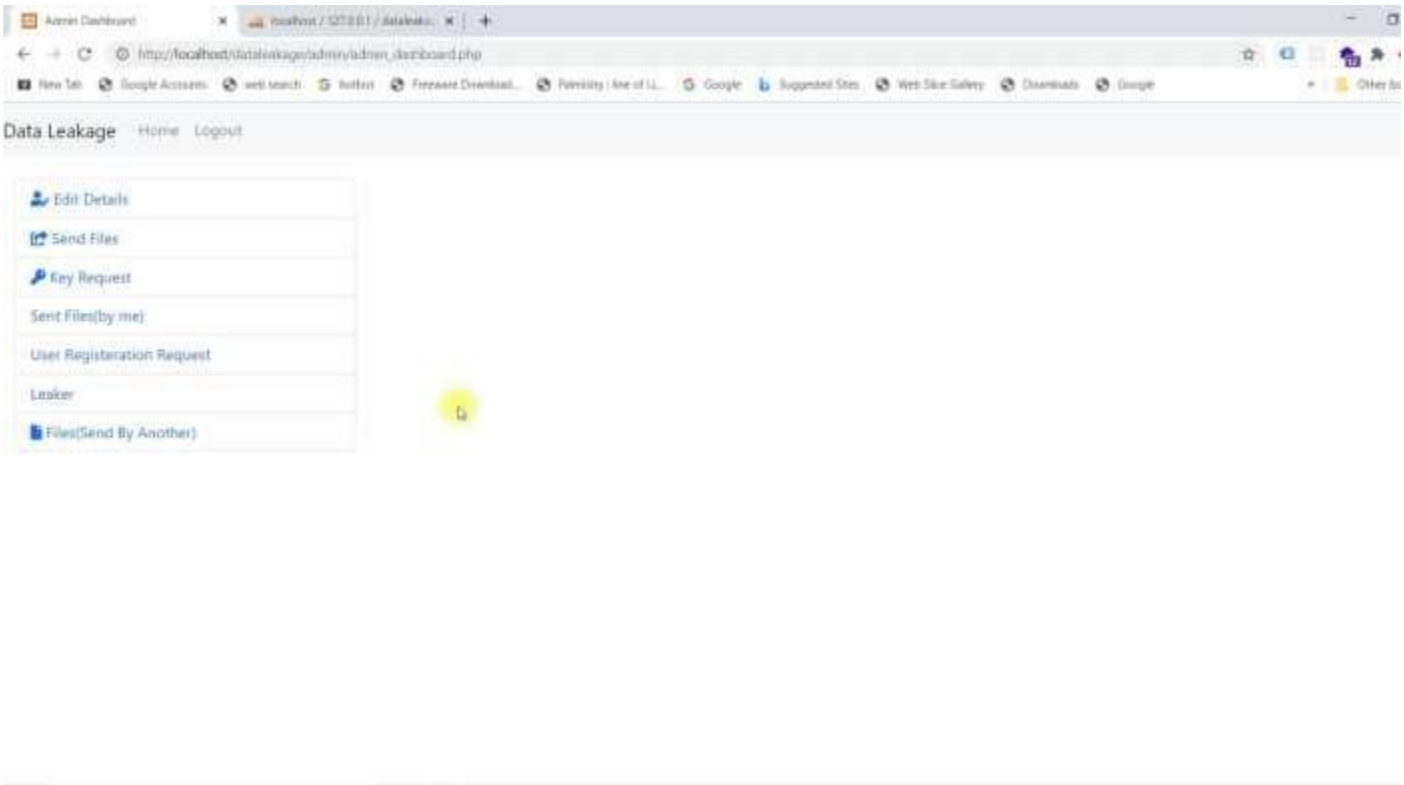
Homepage



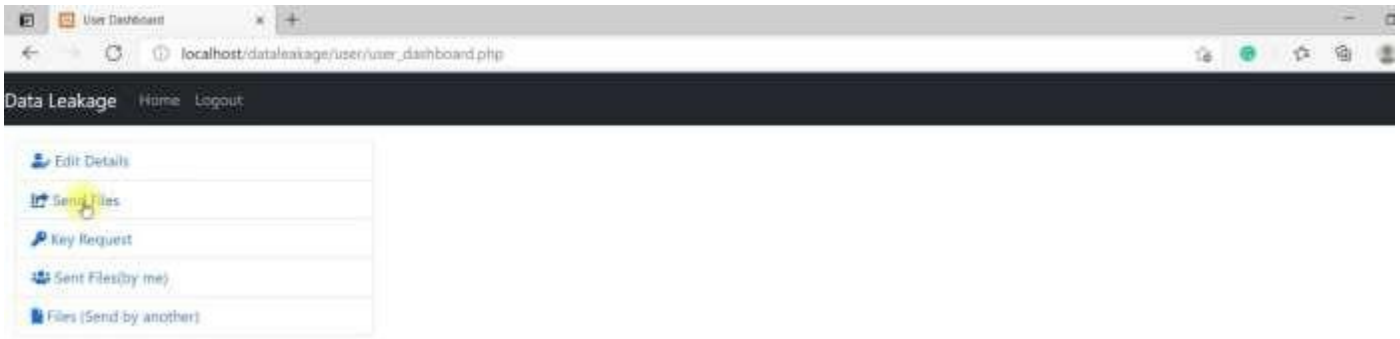
# Login Page


Data Leakage
Home
Register
Login
Email Id
Password
Sign In
Forget Password?
Note: Password may be changed, try to forget password using given email id below
Admin:
email: admin@gmail.com
password: 12345
Distributor:
email: distributor@gmail.com
password: 12345
User:

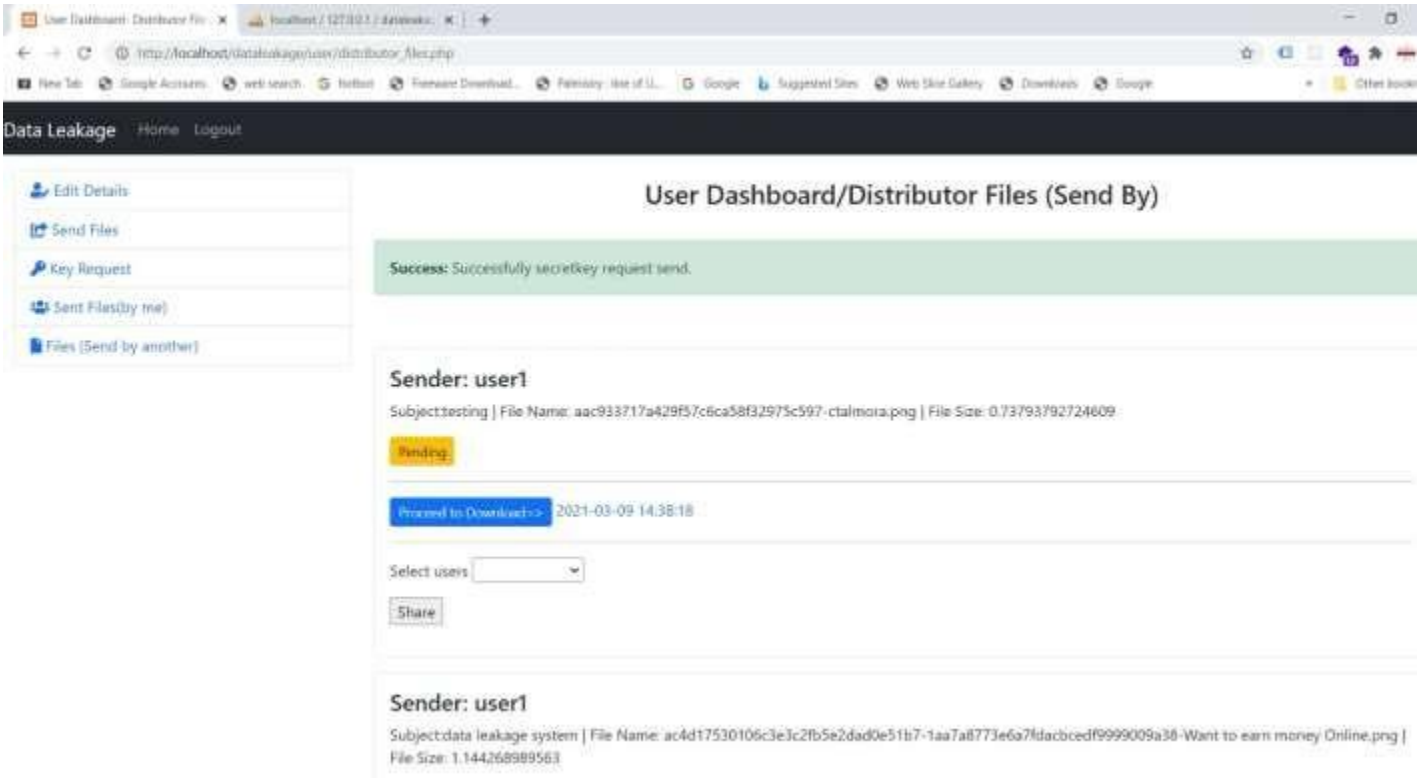
Admin Homepage



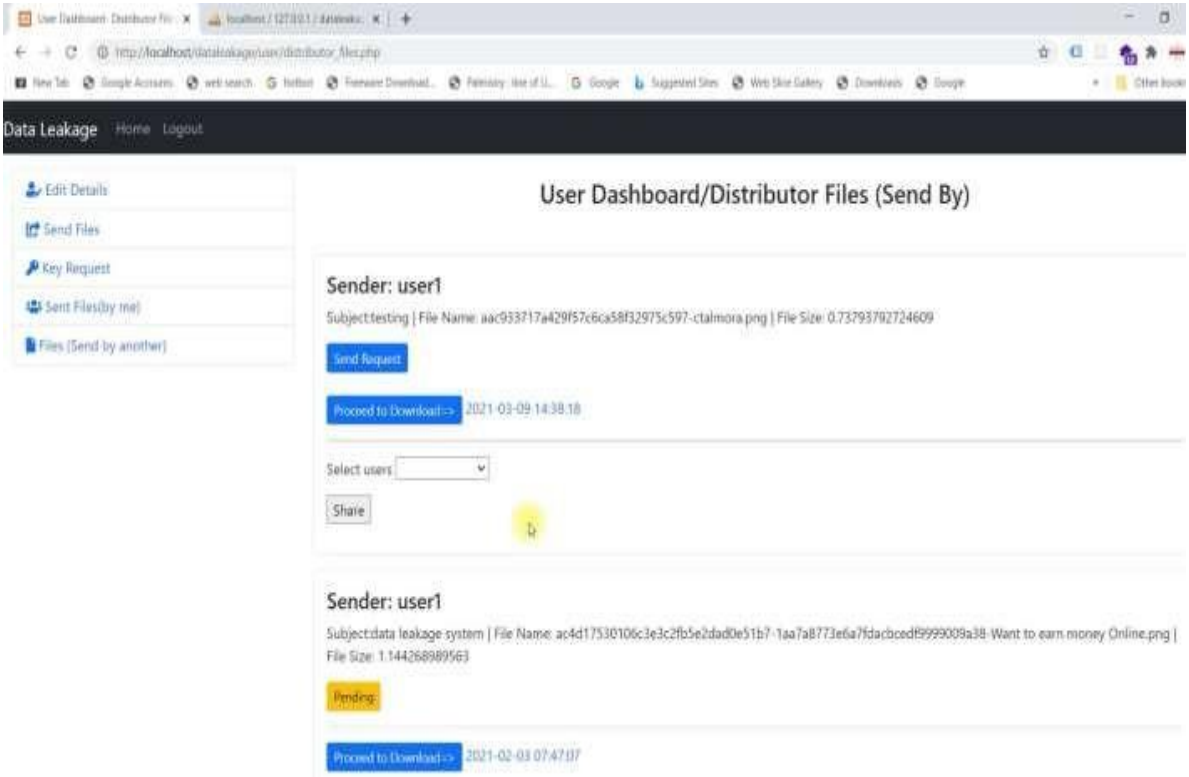
User Homepage

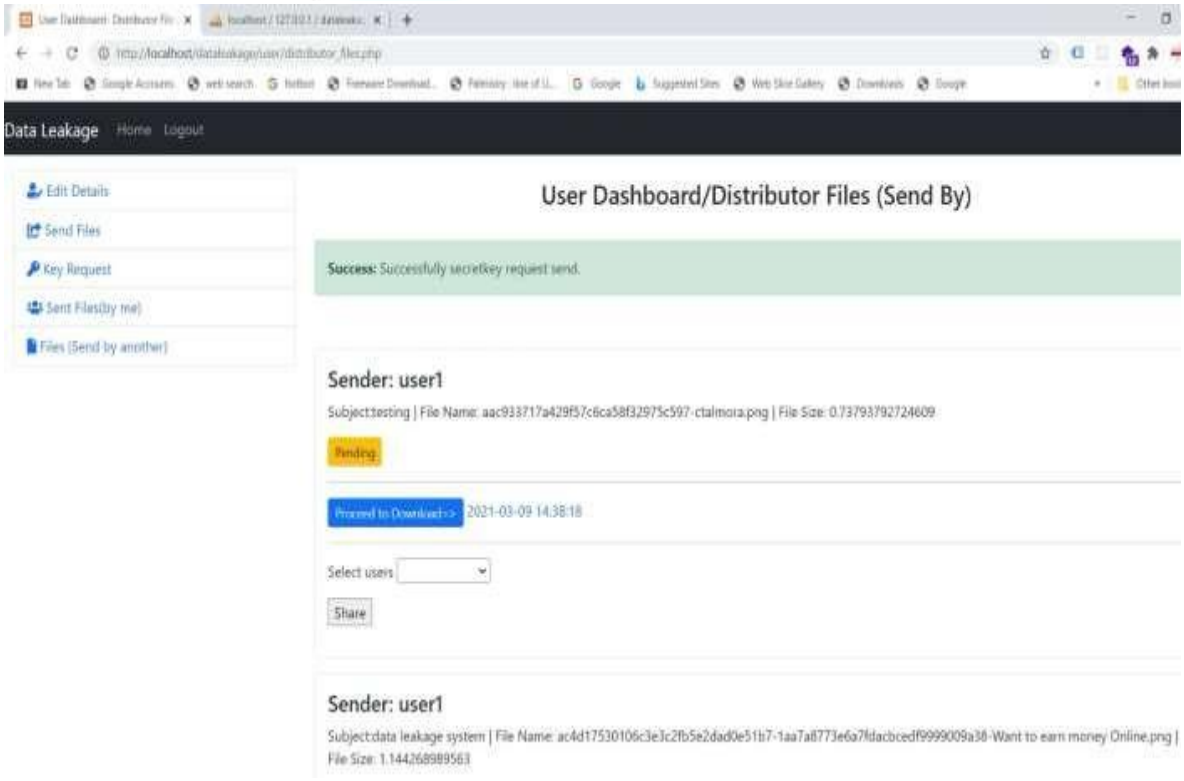


File Sharing



Request page

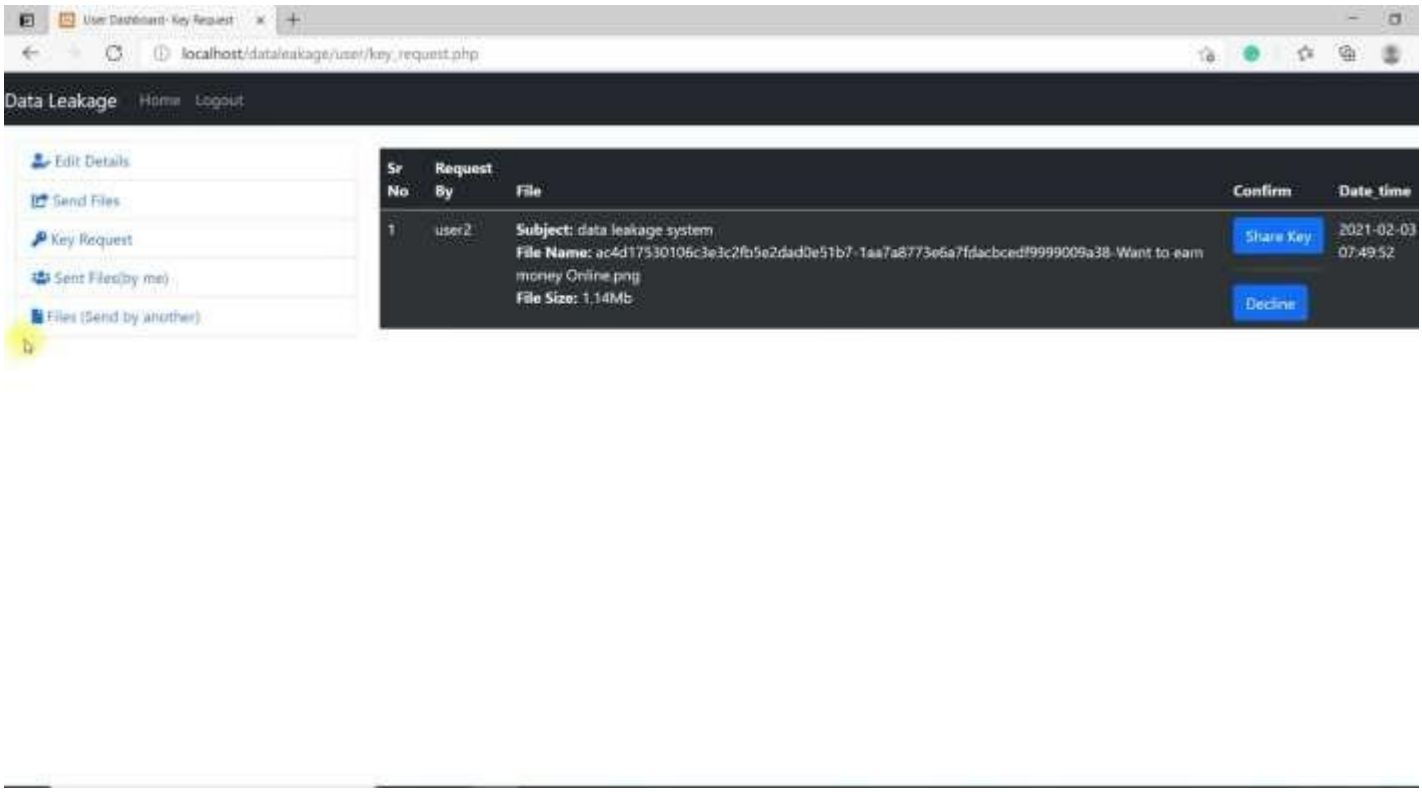


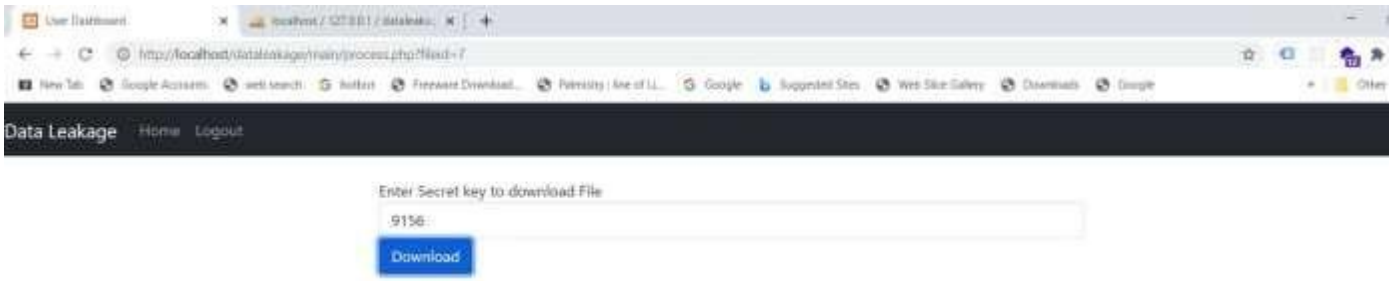
Sender Page


User Dashboard- Key Request
localhost/dataleakage/user/key_request.php
Data Leakage
Home
Logout
Edit Details
Send Files
Key Request
Sent Files(by me)
Files (Send by another)
Sr No
Request By
File
Confirm
Date_time
1
user2
Subject: data leakage system
File Name: ac4d17530106c3e3c2fb5e2dad0e51b7-1aa7a8773e6a7fdacbcedf9999009a38-Want to earn money Online.png
File Size: 1.14Mb
Share Key
Decline
2021-02-03 07:49:52

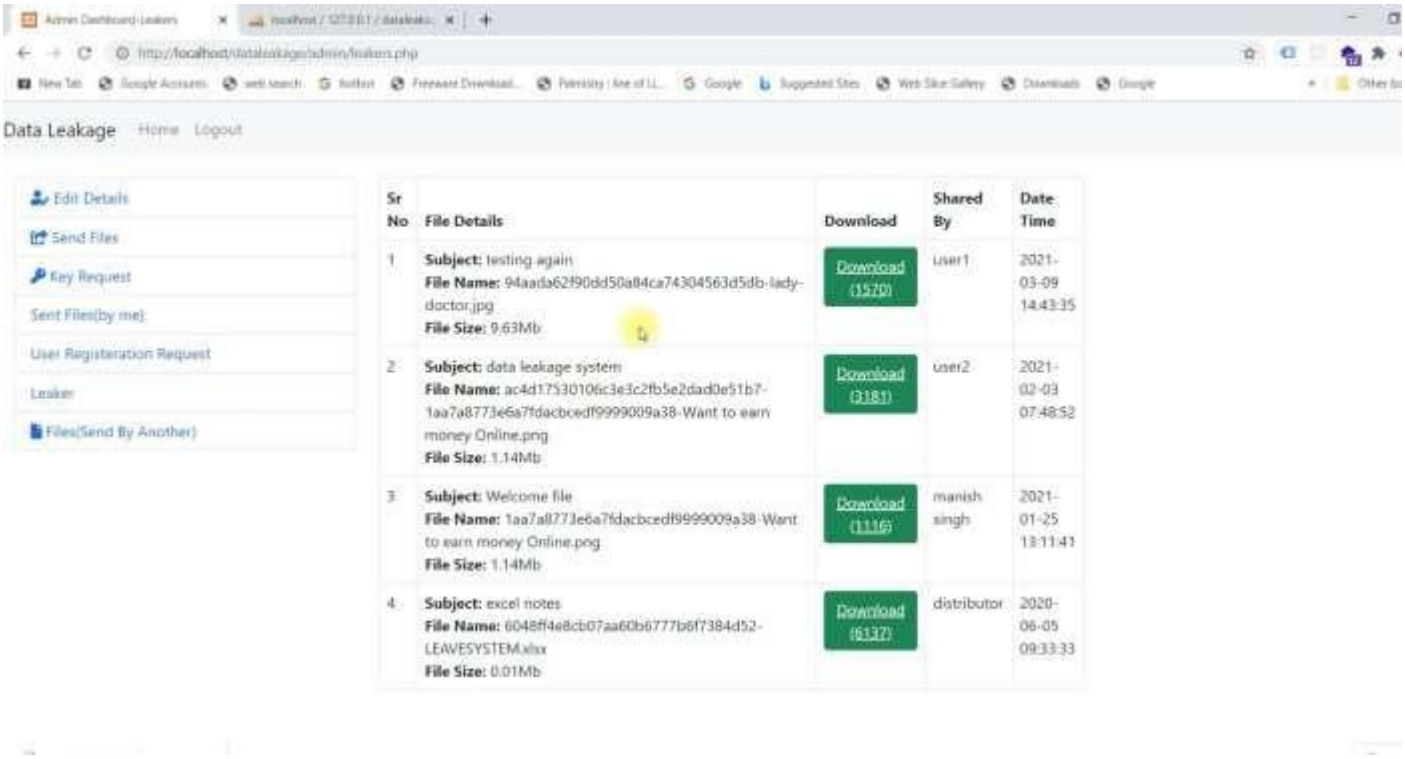
Receiver Page



Enter (Secret Number) Page



Admin page


Data Leakage
Home
Logout
Edit Details
Send Files
Key Request
Sent Files(by me)
User Registeration Request
Leaker
Files(Send By Another)
Sr No
File Details
Download
Shared By
Date Time
1
Subject: testing again
File Name: 94aada62f90dd50a84ca74304563d5db-lady-doctor.jpg
File Size: 9.63Mb
Download (1570)
user1
2021-03-09 14:43:35
2
Subject: data leakage system
File Name: ac4d17530106c3e3c2fb5e2dad0e51b7-1aa7a8773e6a7fdacbcedf9999009a38-Want to earn money Online.png
File Size: 1.14Mb
Download (3181)
user2
2021-02-03 07:48:52
3
Subject: Welcome file
File Name: 1aa7a8773e6a7fdacbcedf9999009a38-Want to earn money Online.png
File Size: 1.14Mb
Download (1116)
manish singh
2021-01-25 13:11:41
4
Subject: excel notes
File Name: 6048ff4e8cb07aa60b6777b6f7384d52-LEAVESYSTEM.xlsx
File Size: 0.01Mb
Download (6137)
distributor
2020-06-05 09:33:33

# SOFTWARE TESTING

## Testing

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

## Black box testing

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

**Specification-based testing:** Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.
Specification-based testing is necessary, but it is insufficient to guard against certain risks.

**Advantages and disadvantages:** The black box tester have no "bonds" with the code, and a tester's perception is very simple: a code must have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers do not. But, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. As a result, there are situations when a tester writes many test cases to check something that could have been tested by only one test case, and/or some parts of the back-end are not tested at all.Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

**White box testing:**

White box testing is when the tester has access to the internal data structures and algorithms including thecode that implement these.

Types of white box testing

The following types of white box testing exist:

- API testing (application programming interface) - Testing of the application using Public and Private APIs.

- Code coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer can createtests to cause all statements in the program to be executed at least once).
- Fault injection methods - improving the coverage of a test by introducing faults to test code paths.
- Mutation testing methods.
- Static testing - White box testing includes all static testing.

## Code completeness evaluation

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested. Two common forms of code coverage are:

- ***Function coverage***, which reports on functions executed.
- ***Statement coverage***, which reports on the number of lines executed to complete the test.

They both return code coverage metric, measured as a percentage.

**Integration Testing**

**Integration testing** is any type of software testing, that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be localized more quickly and fixed.

## Acceptance testing

Acceptance testing can mean one of two things

1. A smoke test is used as an acceptance test prior to introducing a new build to the main testing process, i.e.before integration or regression.
2. Acceptance testing performed by the customer, often in their lab environment on their own HW, is knownas user acceptance testing (UAT).

## Non-Functional Software Testing

Special methods exist to test non-functional aspects of software.

- Performance testing checks to see if the software can handle large quantities of data or users. This is generally referred to as software scalability. This activity of Non-Functional Software Testing is often referred to as Endurance Testing.
- Stability testing checks to see if the software can continuously function well in or above an acceptable period. This activity of Non-Functional Software Testing is oftentimes referred to as load (or endurance) testing.
- Usability testing is needed to check if the user interface is easy to use and understand.
- Security testing is essential for software that processes confidential data to prevent system intrusion by hackers.
- Internationalization and localization is a needed to test these aspects of software, for which a pseudo localization method can be used.

In contrast to functional testing, which establishes the correct operation of the software (correct in that it matches the expected behavior defined in the design requirements), non-functional testing verifies that the software functions properly even when it receives invalid or unexpected inputs. Software fault injection, in the form of fuzzing, is an example of non-functional testing. Non-functional testing, especially for software, is designed to establish whether the device under test can tolerate invalid or unexpected inputs, thereby establishing the robustness of input validation routines as well as error-handling routines. Various commercial non-functional testing tools are linked from the Software fault injection page; there are also numerous open-source and free software tools available that perform non-functional testing.

## Destructive testing

Destructive testing attempts to cause the software or a sub-system to fail, in order to test its robustness. We will be using the black box testing initially to test out the modules and once we are confident of each module working in a cohesive manner, it will be integrated, and integration testing will be performed.

# SYSTEM MAINTENANCE

Maintenance Process finds out essential changes of the market or business or to correct some errorsand tries to implement it in the existing system:

**CORRECTIONS:**

1. Minor changes in the processing logic of system
2. Improving Response time
3. Revisions on formats of data input
4. Improving User interface.

**ADAPTATIONS:**

1. The Proposed system should work on a any device with equal response time as for any greater versiondevice

Achieving greater Scalability and Performance.

# FUTURE ENHANCEMENT

Our future work includes the inquiring of agent guilt models that capture leakage scenarios that are not studied in this paper. For instance, what is the appropriate model for cases where agents can collude and identify fake tuples? Another open problem is the extension of our allocation strategies so that they can handle agent requests in an online fashion (the presented strategies assume that there is a fixed set of agents with requests known in advance).

# CHAPTER 5

## CONCLUSION

In a perfect world there would be no need to handover sensitive data to agents that may unknowingly or maliciously leak it. And even if we had to hand over sensitive data, in a perfect world we could watermark each object so that we could trace its origins with absolute certainty. However, in many cases we must indeed work with agents that may not be 100% trusted, and we may not be certain if a leaked object came from an agent or from some other source, since certain data cannot admit watermarks. In spite of these difficulties, we have shown it is possible to assess the likelihood that an agent is responsible for a leak, based on the overlap of his data with the leaked data and the data of other agents, and based on the probability that objects can be "guessed" by other means. Our model is relatively simple, but we believe it captures the essential trade-offs. The algorithms we have presented implement a variety of data distribution strategies that can improve the distributor's chances of identifying a leaker. We have shown that distributing objects judiciously can make a significant difference in identifying guilty agents, especially in cases where there is large overlap in the data that agents must receive. Our future work includes the investigation of agent guilt models that capture leakage scenarios that are not studied in this paper. For example, what is the appropriate model for cases where agents can collude and identify fake tuples? A preliminary discussion of such model is available in. Another open problem is the extension of our allocation strategies so that they can handle agent requests in an online fashion (the presented strategies assume that there is a fixed set of agents with requests known in advance). Another and most problem to be solved is protecting data before getting leaked.

# REFERENCES


[1] Wu, Jiangjiang, "An Active Data Leakage Prevention Model for Insider Threat Intelligence Information Processing and Trusted Computing" (IPTC), 2011 2nd international Symposium on, 22-23 Oct. 2011.

[2] Papadimitriou, Panagiotis, "A Model for Data Leakage Detection", Data Engineering, 2009. ICDE '09. IEEE 25th International Conference, March 29 2009-April 2 2009.

[3] Xiaosong Zhang , "Research and Application of the Transparent Data Encryption in Intranet Data Leakage Prevention," Computational Intelligence and Security, 2009. CIS '09. International Conference.11-14 Dec. 2009.

[4] Guo, Hongyu, "Identifying and Preventing Data Leakage in Multi-relational Classification," Data Mining Workshops (ICDMW), 2010 IEEE International Conference, 13 Dec,2010.

[5] Bhaduri, Kanishka , "Privacy-Preserving Outlier Detection Through Random Nonlinear Data Distortion," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, Volume: 41, Issue: 1 Feb. 2011.

[6] Li, Xiao-Bai , "A Tree-Based Data Perturbation Approach for Privacy-Preserving Data Mining", Knowledge and Data Engineering, IEEE Transactions, Volume: 18, Issue: 9, 2006.

[7] Lizhong Zhang, Wei Gao; Nan Jiang; "Relational databases watermarking for textual and numerical data", Mechatronic Science, Electric Engineering and Computer (MnEC), 2011 International Conference, 9-22 Aug. 2011.

[8] S. Dominich, "Interaction information retrieval," Journal of Documentation, vol. 50.3, 1994, pp. 197-212, doi: 10.1108/eb026930.

[9] S. Dominich, A. Skrop, and Zs. Tuza, “Formal Theory of Connectionist Web Retrieval,” Soft Computing in Web Information Retrieval, Studies in Fuzziness and Soft Computing, vol. 197, 2006, pp. 163-194.

[10]W. B. Croft, D.Metzler, and T. Strohman, Search engines: Information retrieval in practice (p. 283). Reading: Addison-Wesley, 2010.

[11]Y. Liu, C. Corbett, K. Chiang, R. Archibald, B. Mukherjee, and D. Ghosal, “SIDD: A framework for detecting sensitive data exfiltration by an insider attack,” In System Sciences, 2009, HICSS'09, pp. 1-10, IEEE.

[12] Sachin Sharma; Diksha Sharma; Pankaj Vaidya: “Analytics in Education Using Big Data” Volume: 04,Issue: 11, November 2014

[13] Harshawardhan S. Bhosale, Prof.Devendra P. Gadekar: “A Review Paper on Big Data and Hadoop” in International Journal of Scientific and Research Publications, Volume: 04, Issue: 10, October 2014.

[14] Joseph Grafsgaard; Joseph Wiggins; Kristy Elizabeth Boyer; Eric Wiebe and James Lester: “Predicting learning and affect from multimodal data streams in task-oriented tutorial dialogue”, Proceedings of the7th International Conference on Educational Data Mining, 2014.