DATA LEAKAGE DETECTION

INFORMATION SECURITY MANAGEMENT
J component

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

under the Guidancce of:

SENDHIL KUMAR K.S

Associate Professor
SCOPE

Submitted by

VRINDA CHOPRA (18BCE0785)

RUCHITA DAMODAR (18BCE0916)

GUNESWAR SINGH (18BCE0960)

ASHUTOSH DEVKOTA (18BCE2465)

B.Tech.

In

Computer Science and Engineering

School of Computer Science & Engineering

1. Abstract

1.1. Theoretical Background

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. Data leakage is the unauthorized transmission of data from within an organization to an external destination or recipient. Data loss prevention (DLP) is a strategy that ensures end users do not send confidential or sensitive information outside of the enterprise network.

1.2. Motivation

Data leakage poses a serious issue for companies as the number of incidents and the cost to those experiencing them continue to increase. Data leakage threats usually occur via the web and email, but can also occur via mobile data storage devices such as optical media, USB keys, and laptops. Data leakage is enhanced by the fact that transmitted data including emails, instant messaging, website forms, and file transfers among others, are largely unregulated and unmonitored on their way to their destinations. Therefore it is important to detect data leakage and prevent it successfully.

1.3. Aim of the proposed Work

The main aim of this project is to provide complete information about the data/content that is accessed by the users within the website. Forms Authentication technique is used to provide security to the website to prevent the leakage of the data.

1.4. Objective(s) of the proposed work

Our objective 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 detecting leakage of a set of objects or records. In this section we develop a model for assessing the "guilt" of agents. 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 agents. In a sense, the fake objects act 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 the distributor can be more confident that agent was guilty.

2. Literature Survey

2.1. Survey of Existing models/ work

Traditionally, leakage detection is handled by watermarking, e.g., a unique code is embedded in each distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious. *E.g.* 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 must be given to various other companies. We call the owner of the data the distributor and the supposedly trusted third parties the agents.

2.2. Summary/Gaps identified in the Survey

In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties. 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 must be given to various other companies. We call the owner of the data the distributor and the supposedly trusted third parties the agents.

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 consider applications where the original sensitive data cannot be perturbed. Perturbation is a very useful technique where the data is modified and made "less sensitive" before being handed to agents. For example, one can add random noise to certain attributes, or one can replace exact values by ranges However, in some cases it is important not to alter the original distributor's data.

For example, if an outsourcer is doing our payroll, he must have the exact salary and customer bank account numbers. If medical researchers will be treating patients (as opposed to simply computing statistics), they may need accurate data for the patients. Traditionally, leakage detection is handled by watermarking, e.g., a unique code is embedded in each distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious.

3. Overview of the Proposed System

3.1. Introduction and Related Concepts

The guilt detection approach we present is related to the data provenance problem: tracing the lineage of an S object implies essentially the detection of the guilty agents. It provides a good overview on the research conducted in this field. Suggested solutions are domain specific, such as lineage tracing for data Warehouses, and assume some prior knowledge on the way a data view is created out of data sources. Our problem formulation with objects and sets is more general and simplifies lineage tracing, since we do not consider any data transformation from Ri sets to S. As far as the data allocation strategies are concerned, our work is mostly relevant to watermarking that is used as a means of establishing original ownership of distributed objects. Watermarks were initially used in images, video and audio data whose digital representation includes considerable redundancy. Our approach and watermarking are similar in the sense of providing agents with some kind of receiver identifying information. However, by its very nature, a watermark modifies the item being watermarked. If the object to be watermarked cannot be modified, then a watermark cannot be inserted. In such cases, methods that attach watermarks to the distributed data are not applicable.

Finally, there are also lots of other works on mechanisms that allow only authorized users to access sensitive data through access control policies. Such approaches prevent in some sense data leakage by sharing information only with trusted parties. However, these policies are restrictive and may make it impossible to satisfy agents' requests.

3.2. Framework, Architecture & Module for the Proposed System (with explanation)

Data Leakage Detection Project propose data allocation strategies that improve the probability of identifying leakages. In some cases, we can also inject "realistic but fake" data records to further improve our chances of detecting leakage and identifying

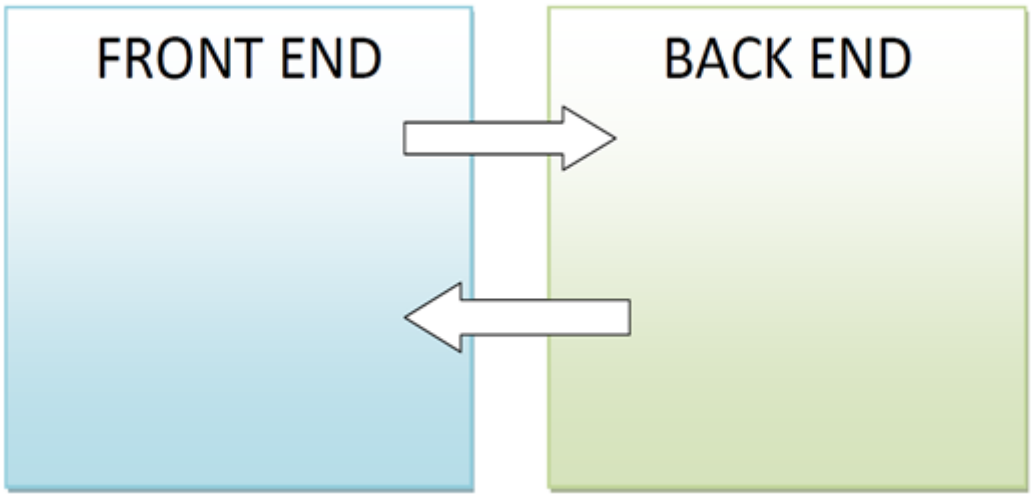
the guilty party. In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties. Another enterprise may out source its data processing, so data must be given to various other companies. There always remains a risk of data getting leaked from the agent. Leakage detection is handled by watermarking, e.g., a unique code is embedded in each distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. But again it requires code modification. Watermarks can sometimes be destroyed if the data recipient is malicious. Traditionally, leakage detection is handled by watermarking, e.g, a unique code is embedded in each distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious.

In this project, we use unobtrusive techniques for detecting leakage of a set of objects or records. Specifically, the following scenario: After giving a set of objects to agents, the distributor discovers some of those same objects in an unauthorized place. (For example, the data may be found on a website, or may be obtained through a legal discovery process.) At this point, the distributor can assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. Using an analogy with cookies stolen from a cookie jar, if we catch Freddie with a single cookie, he can argue that a friend gave him the cookie. But if we catch Freddie with five cookies, it will be much harder for him to argue that his hands were not in the cookie jar. If the distributor sees “enough evidence” that an agent leaked data, he may stop doing business with him, or may initiate legal proceedings. In this project, we develop a model for assessing the “guilt” of agents.

We also used 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 agents. In a sense, the fake objects act as a type of watermark for the entire set, without modifying any individual members. If it turns out that an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty.

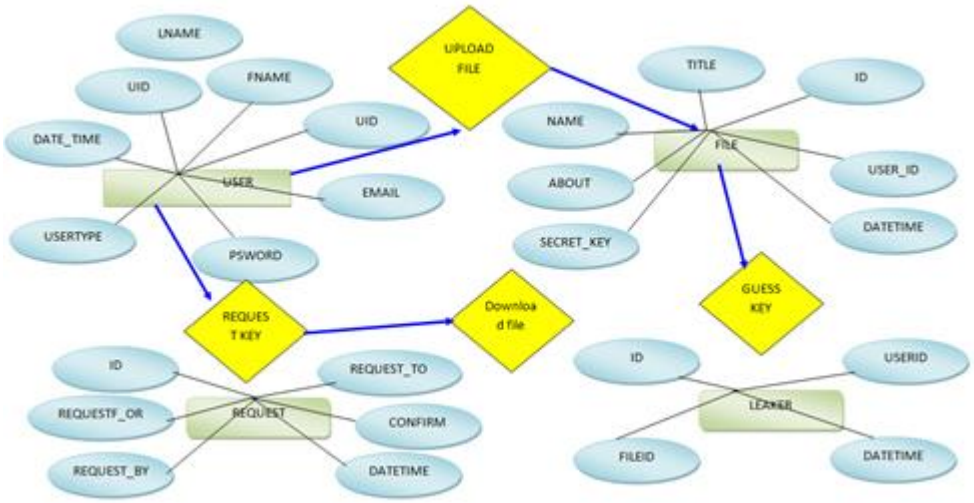
3.3. Proposed System Model (ER Diagram/UML Diagram/Mathematical Modeling)

SYSTEM MODEL

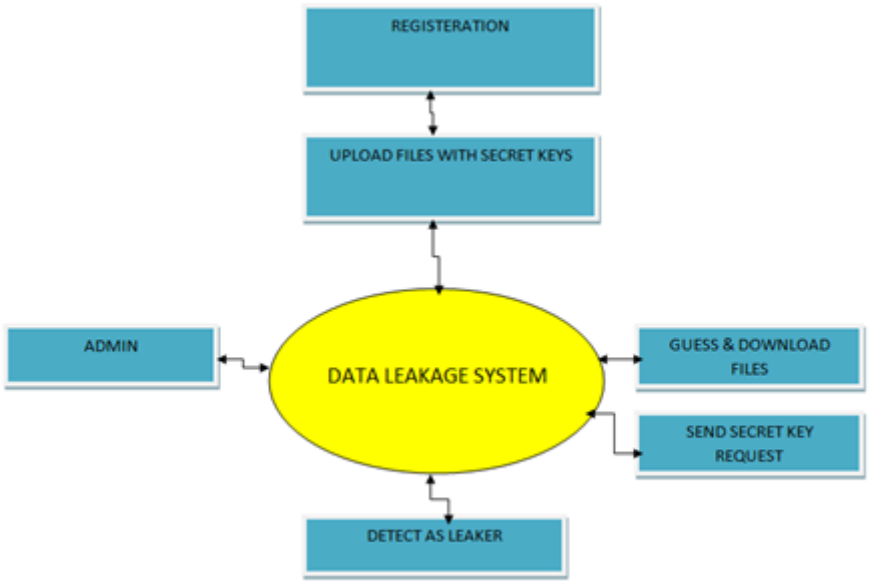


SYSTEM DESIGN

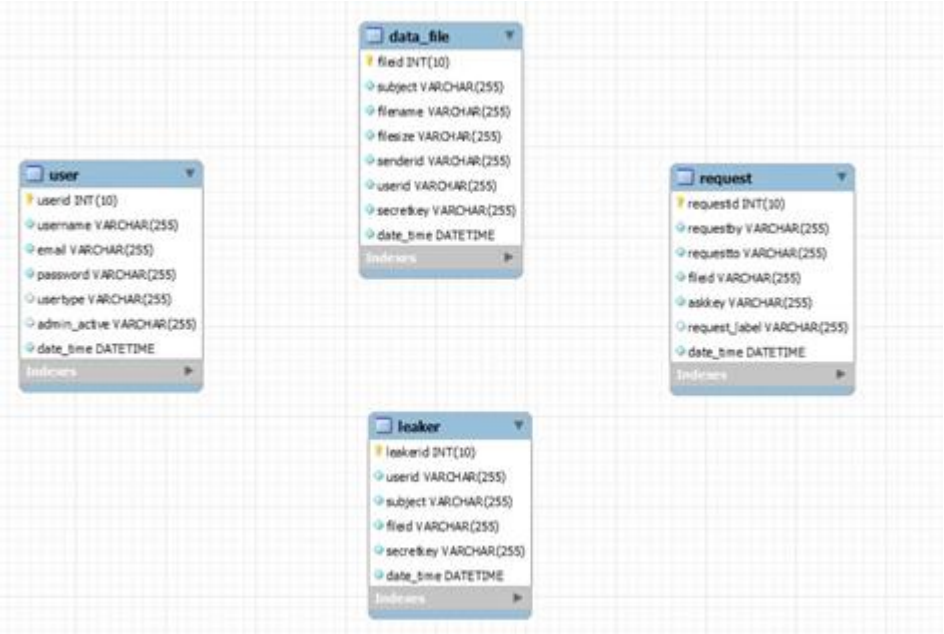
ER DIAGRAM



DFD DIAGRAM



DATABASE



4. Proposed system analysis and design

4.1 Introduction

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 detecting leakage of a set of objects or records. In this section we develop a model for assessing the "guilt" of agents. 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 agents. If it turns out an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty.

To detect data leakage, the method used is that whenever someone user share files with others there will be a secret key. If someone want to download or access file, they need a

secret key which will be request to sender who share this file. After sharing secret key user can download that file. If someone download that file without asking secret key using guess method that will be notified as leaker.

4.2.1 Functional Requirements

4.2.1.1 Product perspective

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.

To detect data leakage, the method used is that whenever someone user share files with others there will be a secret key. If someone want to download or access file, they need a secret key which will be request to sender who share this file. After sharing secret key user can download that file. If someone download that file without asking secret key using guess method that will be notified as leaker.

4.2.1.2 Product features

The website provides features to send and receive important files in a secured manner which prevents and detects data leakage through the following method:

- The website provides different interfaces for admin and other guest users.
- The admin interface provide feature to send files to concerned user.
- Once the user receives the file, he has to request for the key from sender to download it.
- Once the admin gets request for key he can either share the key or decline it.
- The receiver can use that key to download the file.

- But if the key is not shared through legitimate means and the user somehow got hold of the key, still he cannot use the key to download the file. This is how data leakage is prevented.
- And that user get updated in leaker list in admin interface with all details like date and time and admin can choose to deactivate the account of the leaker user. This is how data leakage is detected.

4.2.1.3 User characteristics

In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties. Another enterprise may outsource its data processing, so data must be given to various other companies. There always remains a risk of data getting leaked from the agent. Therefore users of our product are members of such enterprises who are involved in sharing of sensitive data with other parties under a risk of that data getting leaked by some malicious agents.

4.2.1.4 Assumptions and dependencies

We have assumed that our project is for a particular company and has all the dependencies for the same. We have assumed that the admin has all the control and the data can be leaked only by the user. The user is assumed to be verified by the admin and hence no provision for the same is provided.

4.2.1.5 Domain requirements

We required that an internet connection and the admin will be functional after that. We have assumed that the admin has full control for detecting the leaker and disabiling his account. We have also assumed that the admin would give previledges only to verified users. Domain requirements are the requirements which are characteristic of a particular category or domain of projects. The basic functions that a system of a specific domain must necessarily exhibit come under this category.

4.2.1.6 User requirements

The user requirements would be a good internet connection and the availability of better functionalities. The requirements would be an easily available system or device where internet could be accessed and phone connectivity for verification.

4.3 Operational Requirements

4.3.1 Economic

Operational Requirements are capabilities, performance measurement (Measures of Effectiveness, Measures of Performance, Measures of Suitability & Technical Performance Measurements), and processes needed to address mission area deficiencies, evolving threats, emerging technologies, or weapon system cost improvements.

In this project, capabilities are being able to detect and prevent data leakage in the system. Although it's performance is excellent as measured during our test runs. It may be a fact that deficiencies revolve around integrating this system directly into different systems, newer technologies can be used for future development such as crypto for better encryption protection of data.

4.3.2 Environmental

Environmental Requirements are factors that take into effect the effect it has on the environment or vice-versa, and if it is affected by any environmental law, any environmental permit, etc.

Considering this project, environmental costs are minimum and is viable as no risk to environment is caused as compared to blockchain rigs, and cloud servers. The system is completely simple, and very easy to use, can be used using simple electronic devices such as a personal desktop or a laptop.

4.3.3 Social

Social Requirements take into effect any social laws, social permits, basic terms and conditions of employment, core labour standards and IFC Performance Standards that may be a part of the agreement signed for a production to function.

The following project doesn't objects any of the above permits, laws or agreements, protecting the data of the users as it is.

4.3.4 Political

Political requirements are such that in areas where a certain laws and permits allow for a permitted workforce, effect on the environment have to be taken into effect.

All political requirements are satisfied as data kept safe and secure and not leaked, in which case we can identify that particular user and report and block him/her.

4.3.5 Ethically

Ethical requirements are ones which are required for the sake of ethical question and viability of the product.

In fact, the above mentioned product is helpful in such a way to prevent and protect in case of data transfer done safely and securely.

4.3.6 Health and Safety

Health and safety for each user is protected from online scavengers and threat. Users id's and information is made secure in a way that it can't be linked back to a particular person if a user attempts to do so. Only admin has the access to a user's information and that too is also a secure transaction.

4.3.7 Sustainability

Sustainability is a very important aspect as it requires a lot of research and thought for a product to be able to adapt and change according to the varying needs of the users at this platform.

4.3.7 Legality

The legality aspect comes from various laws, permits and others which prevent and protect users, their information, and their data which is kept safe with provisions to detect and prevent data leakages and identify the leaker.

4.3.7 Inspectability

The inspectibility aspect of the product comes into effect when and if the following product in question is required for a inspection into it’s functioning and data collection policies.

In such a case, the following project is free to download and inspect as it is available on Github repository.

4.2.4. H/W Requirements(details about Application Specific Hardware)

Computer System:

2 gb ram, 500 gb hard disk, internet

4.2.5. S/W Requirements(details about Application Specific Software)

Front End technology

Html5, css3, javascript , bootstrap4.

Backend technology

Php, mysql

OS

Linux or window

Code editor

Notepad, notepad++, sublime, visual code, atom

Web browser

Google chrome, Mozilla firefox, Microsoft edge

5. Implementation

- **ADMIN**

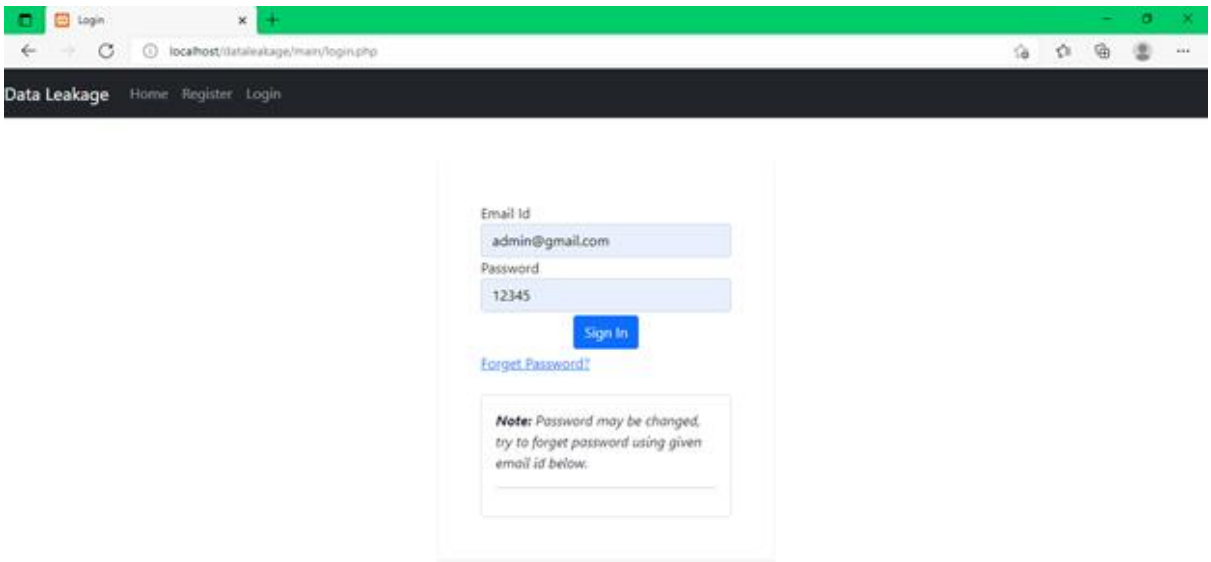


Fig 1: The admin login

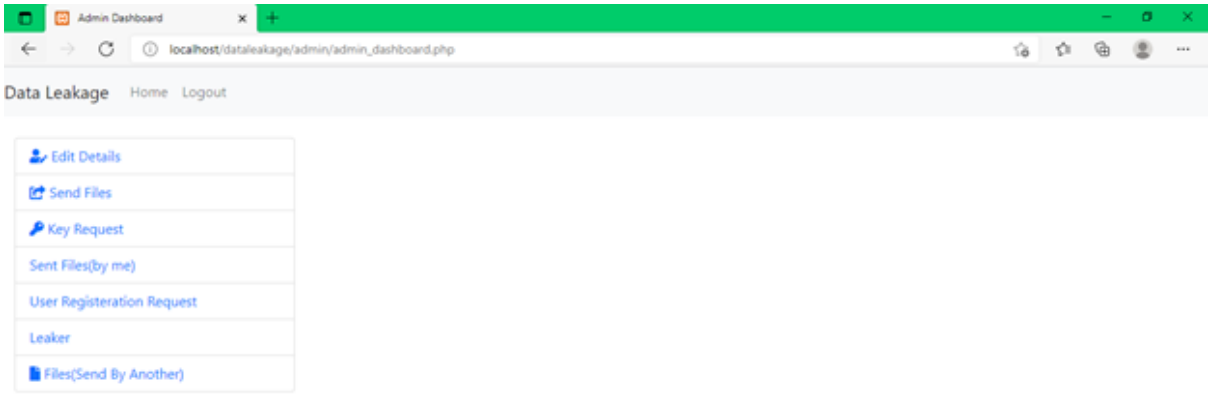


Fig 2: The admin dashboard

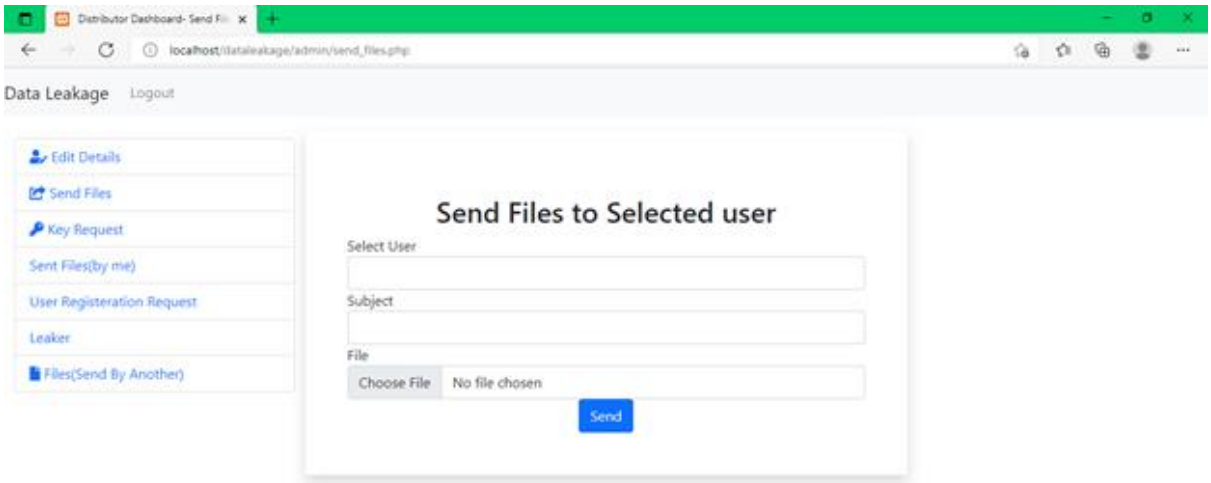


Fig 3: The admin interface provide feature to send files to concerned user.

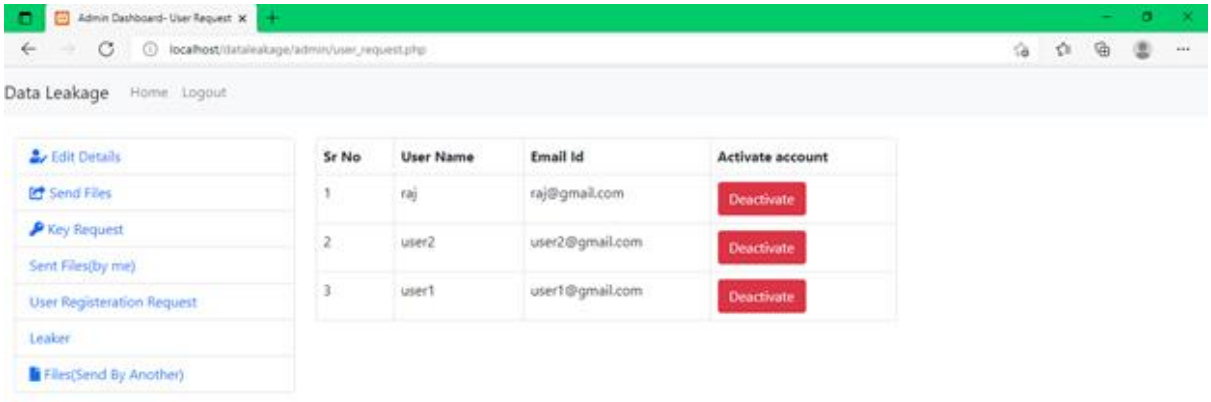


Fig 4: admin can choose to deactivate the account of the leaker user.

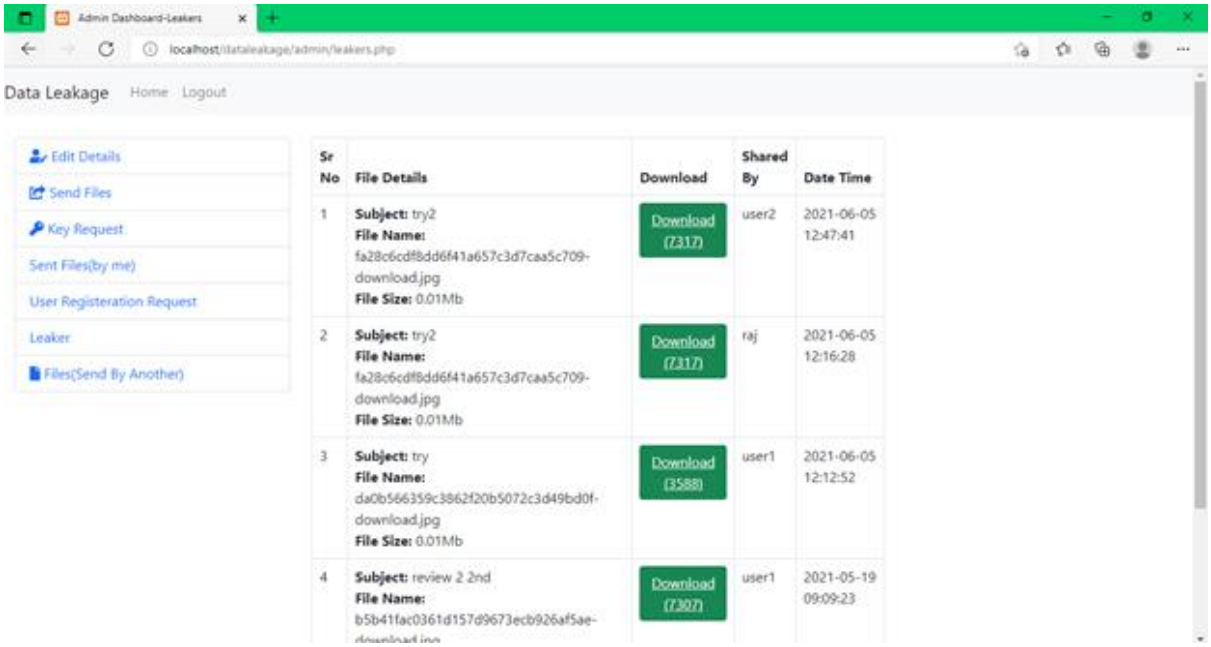


Fig 5: leaker user get updated in leaker list in admin interface with all details like date and time

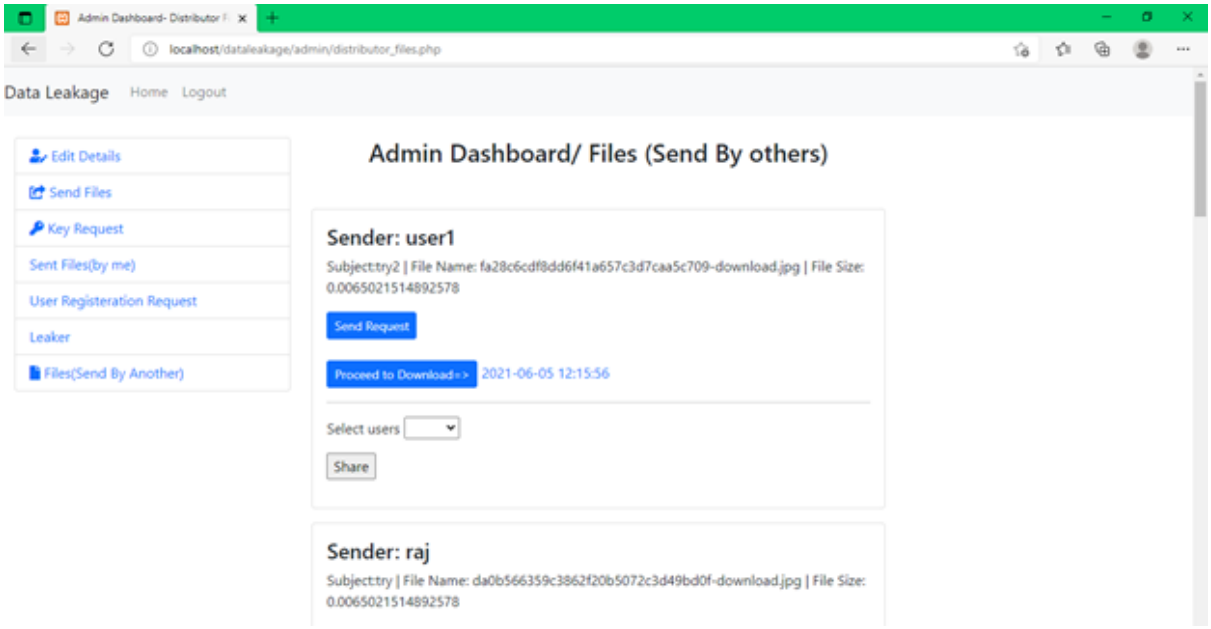


Fig 6: Once the user receives the file, he has to request for the key from sender to download it.

- **GUEST**

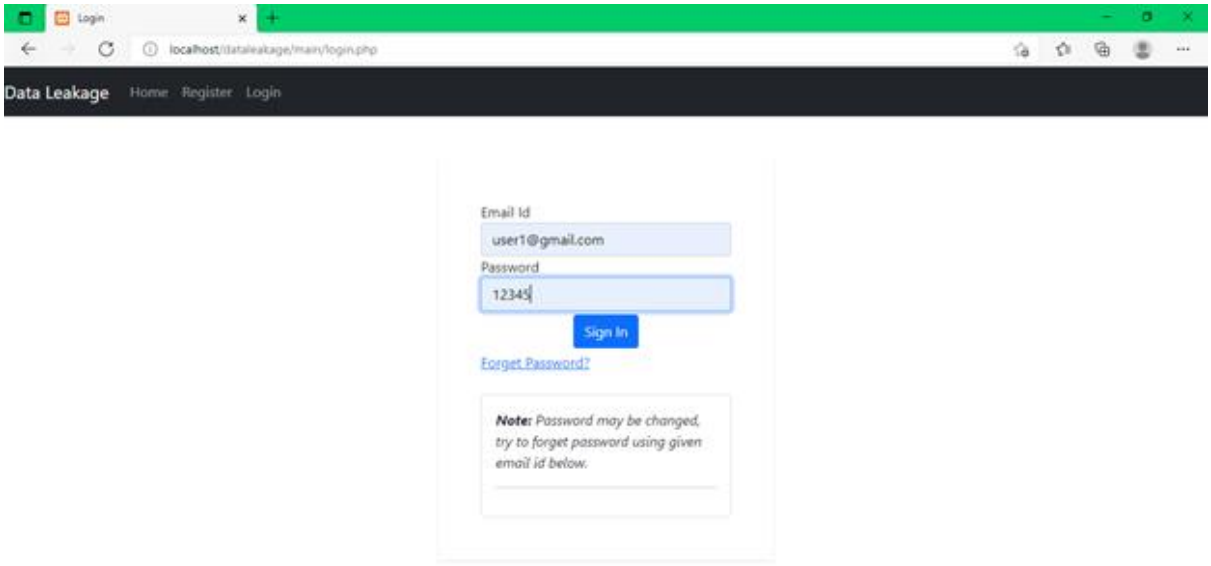


Fig 7: The guest user login

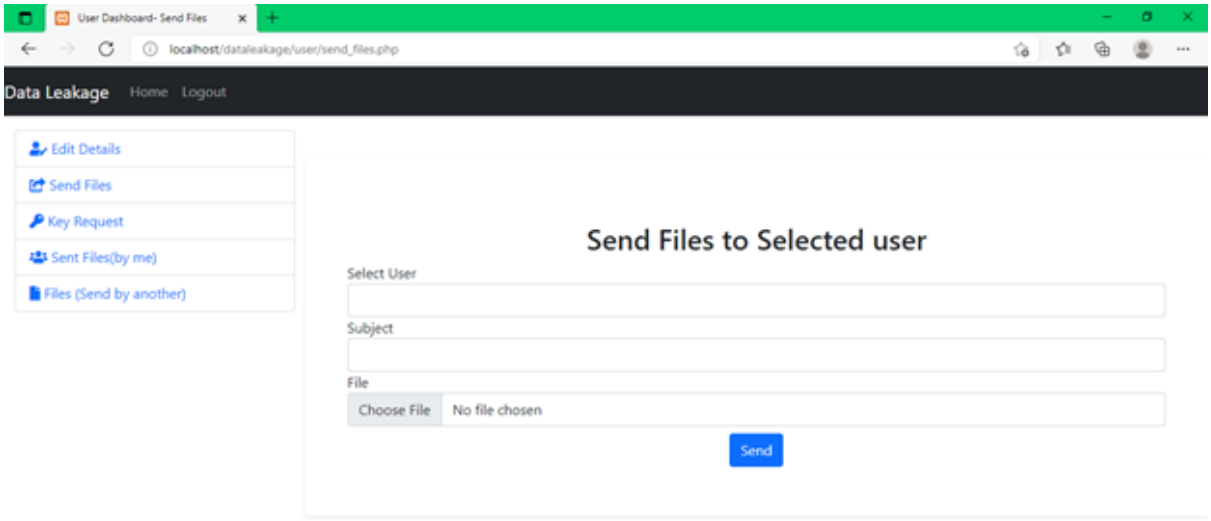


Fig 8: The interface provide feature to send files to concerned user.

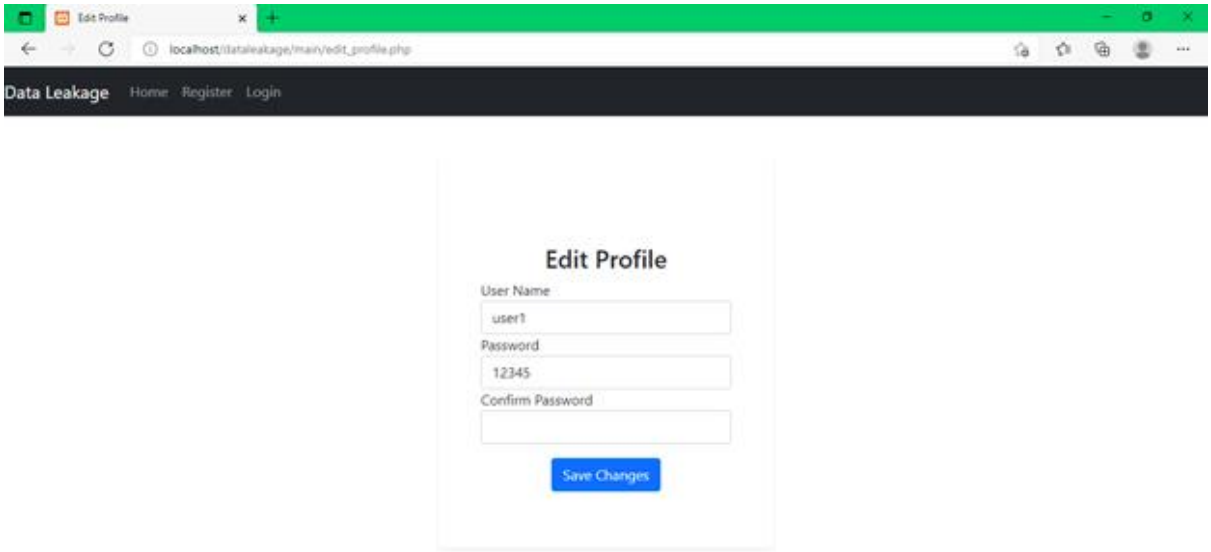


Fig 9: edit profile feature

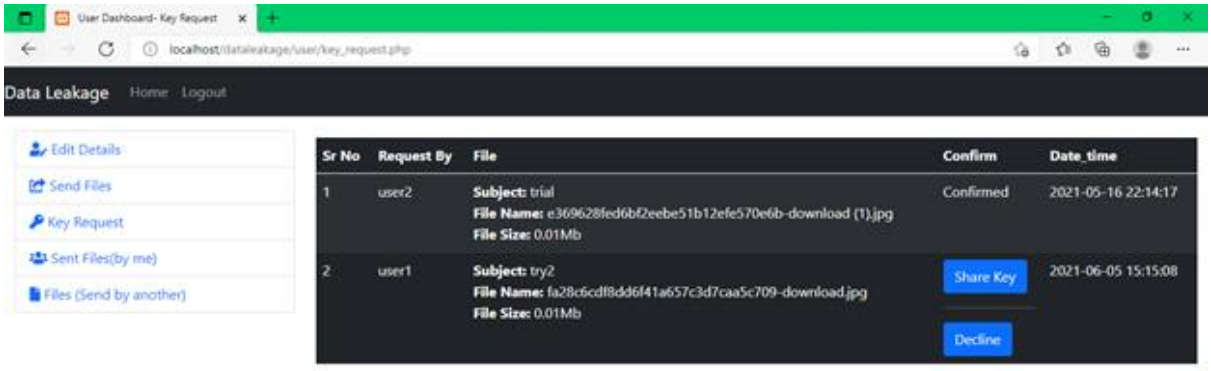


Fig 10: Once the admin gets request for key he can either share the key or decline it.

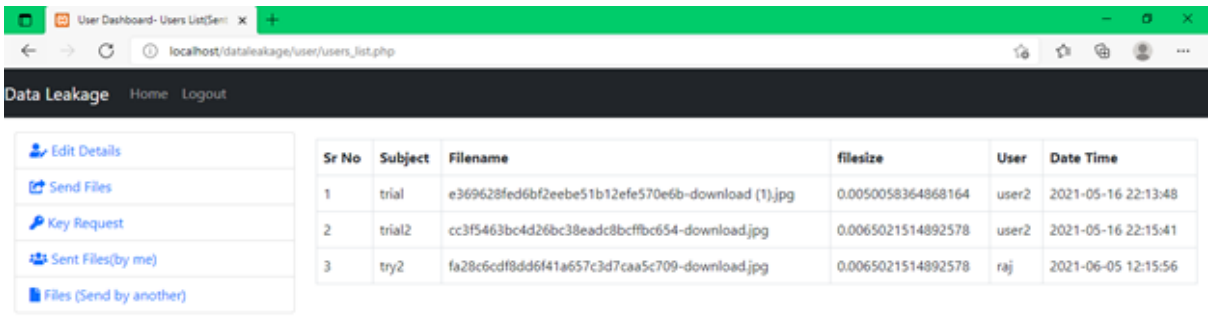


Fig 11: the users list and details of the files shared

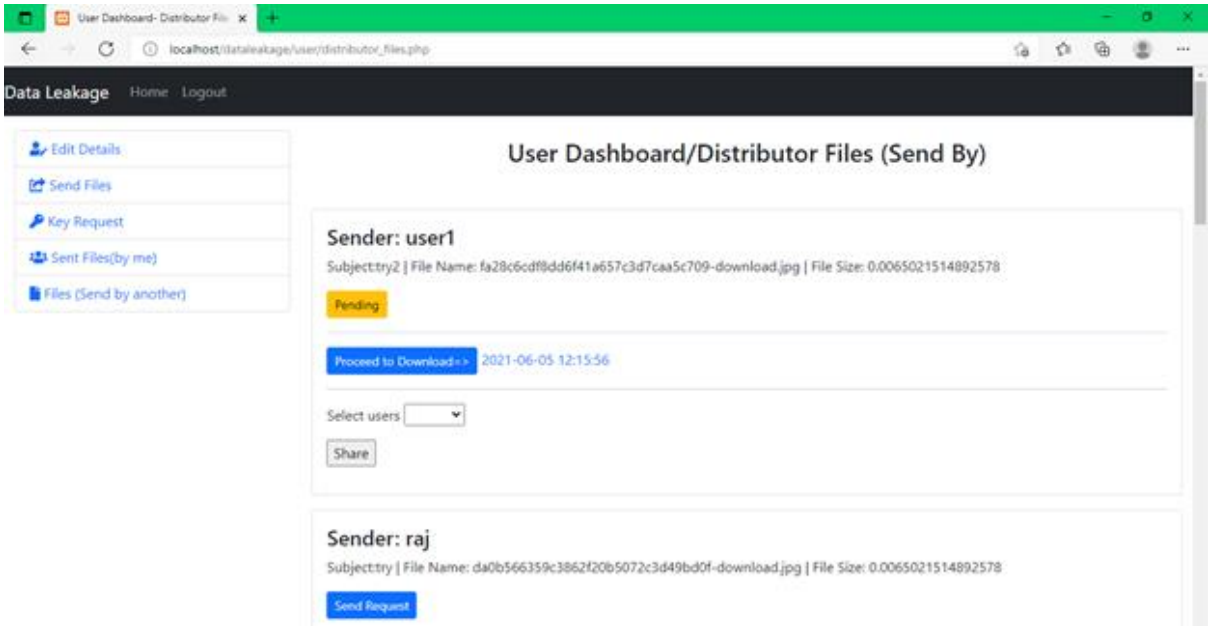
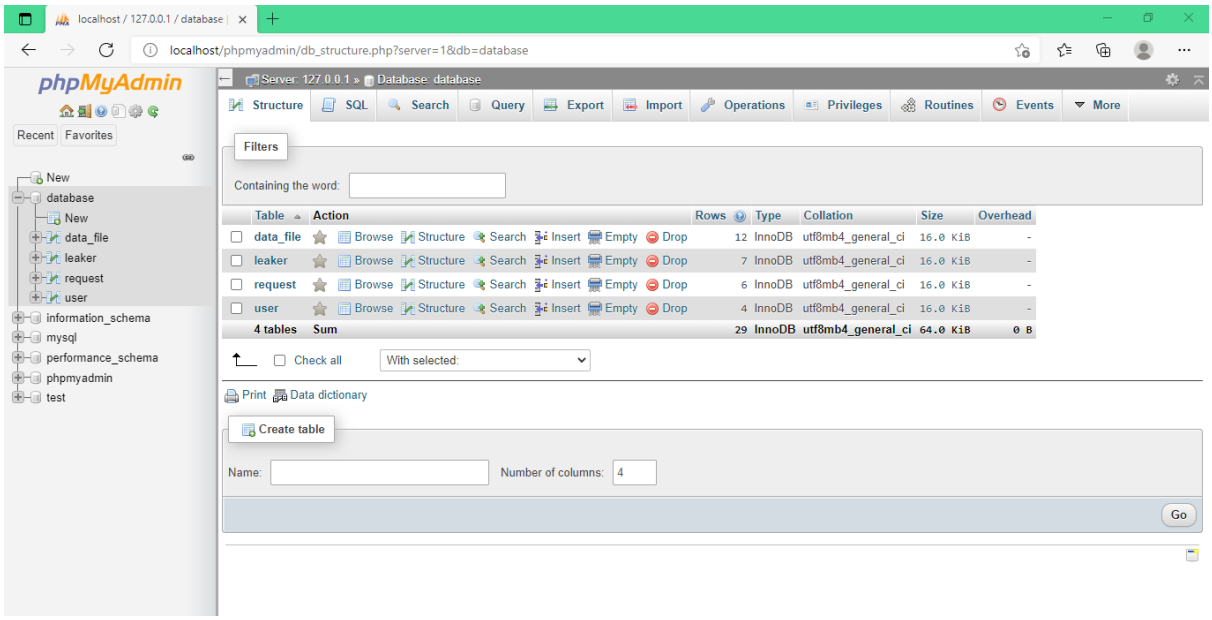
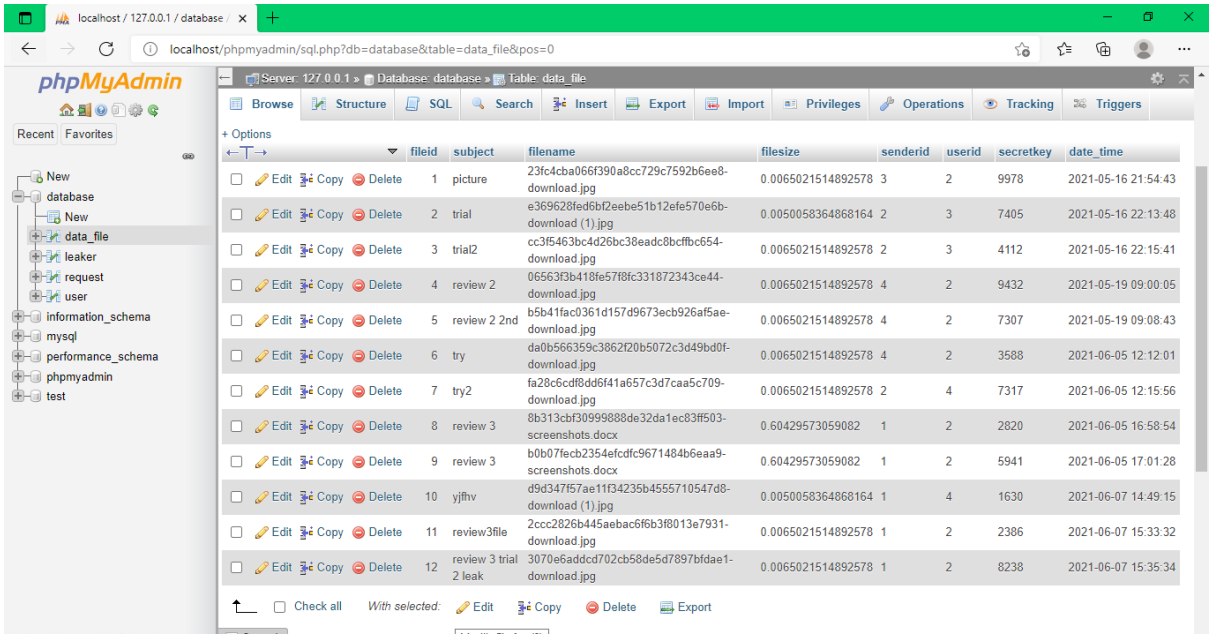
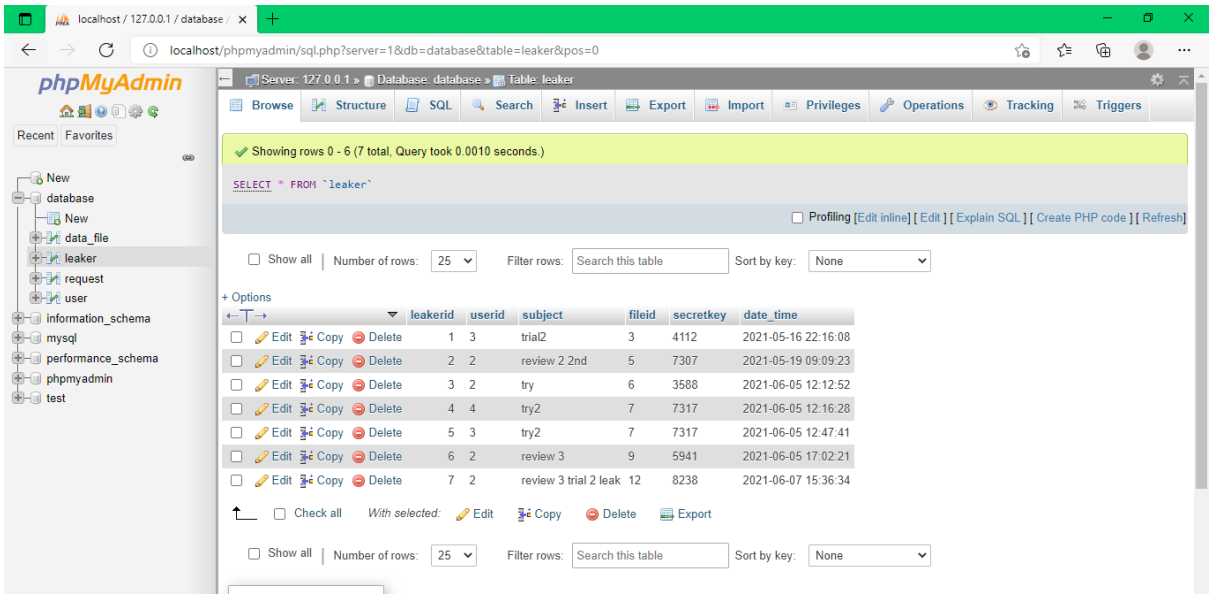


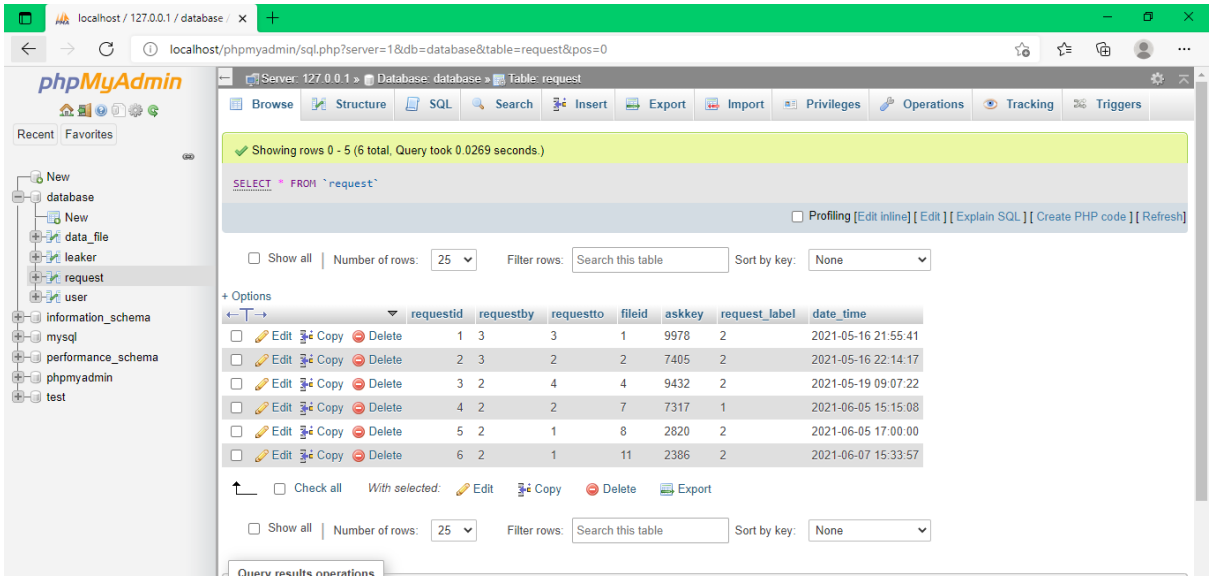
Fig 12: Once the user receives the file, he has to request for the key from sender to download it.

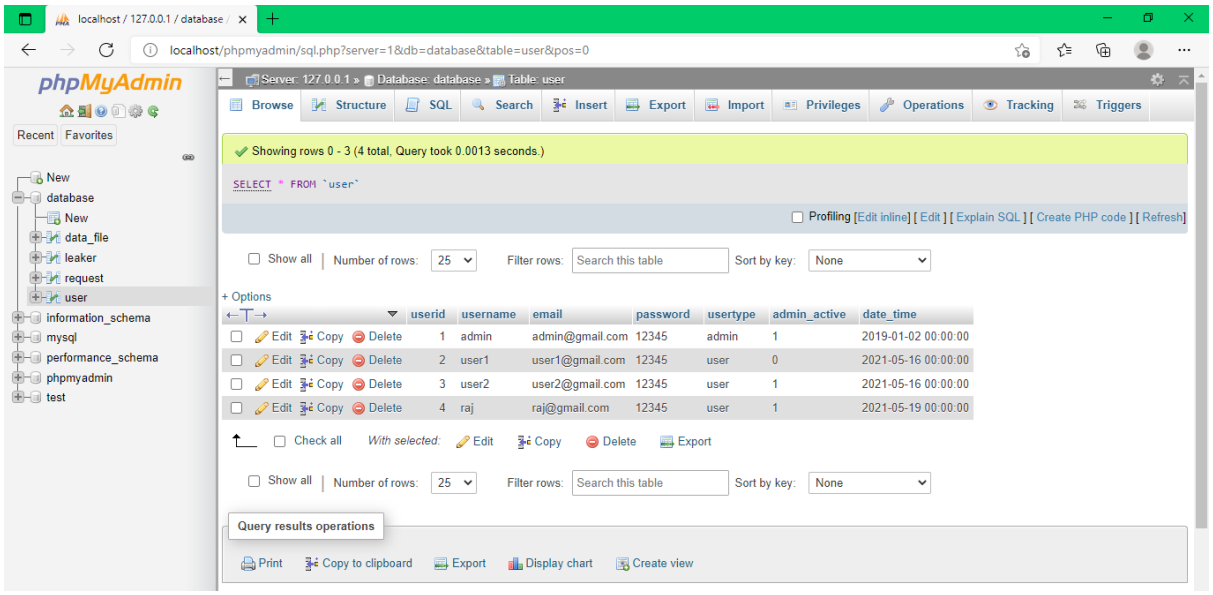
DATABASE IMAGES




localhost / 127.0.0.1 / database
localhost/phpmyadmin/sql.php?db=database&table=data_file&pos=0
phpMyAdmin
Recent Favorites
New
database
New
data_file
leaker
request
user
information_schema
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: database » Table: data_file
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
+ Options
fileid subject filename filesize senderid userid secretkey date_time
1 picture 23fc4cba066f390a8cc729c7592b6ee8-download.jpg 0.0065021514892578 3 2 9978 2021-05-16 21:54:43
2 trial e369628fed6bf2eebe51b12efe570e6b-download (1).jpg 0.0050058364868164 2 3 7405 2021-05-16 22:13:48
3 trial2 cc3f5463bc4d26bc38eadc8bcffbc654-download.jpg 0.0065021514892578 2 3 4112 2021-05-16 22:15:41
4 review 2 06563f3b418fe57f8fc331872343ce44-download.jpg 0.0065021514892578 4 2 9432 2021-05-19 09:00:05
5 review 2 2nd b5b41fac0361d157d9673ecb926af5ae-download.jpg 0.0065021514892578 4 2 7307 2021-05-19 09:08:43
6 try da0b566359c3862f20b5072c3d49bd0f-download.jpg 0.0065021514892578 4 2 3588 2021-06-05 12:12:01
7 try2 fa28c6cdf8dd6f41a657c3d7caa5c709-download.jpg 0.0065021514892578 2 4 7317 2021-06-05 12:15:56
8 review 3 8b313cbf30999888de32da1ec83ff503-screenshots.docx 0.60429573059082 1 2 2820 2021-06-05 16:58:54
9 review 3 b0b07fecb2354efcdfc9671484b6eaa9-screenshots.docx 0.60429573059082 1 2 5941 2021-06-05 17:01:28
10 yjfhv d9d347f57ae11f34235b4555710547d8-download (1).jpg 0.0050058364868164 1 4 1630 2021-06-07 14:49:15
11 review3file 2ccc2826b445aebac6f6b3f8013e7931-download.jpg 0.0065021514892578 1 2 2386 2021-06-07 15:33:32
12 review 3 trial 2 leak 3070e6addcd702cb58de5d7897bfdae1-download.jpg 0.0065021514892578 1 2 8238 2021-06-07 15:35:34
Check all With selected: Edit Copy Delete Export


localhost / 127.0.0.1 / database
localhost/phpmyadmin/sql.php?server=1&db=database&table=leaker&pos=0
phpMyAdmin
Recent Favorites
New
database
New
data_file
leaker
request
user
information_schema
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: database » Table: leaker
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Showing rows 0 - 6 (7 total, Query took 0.0010 seconds.)
SELECT * FROM `leaker`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options
leakerid userid subject fileid secretkey date_time
1 3 trial2 3 4112 2021-05-16 22:16:08
2 2 review 2 2nd 5 7307 2021-05-19 09:09:23
3 2 try 6 3588 2021-06-05 12:12:52
4 4 try2 7 7317 2021-06-05 12:16:28
5 3 try2 7 7317 2021-06-05 12:47:41
6 2 review 3 9 5941 2021-06-05 17:02:21
7 2 review 3 trial 2 leak 12 8238 2021-06-07 15:36:34
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None





6. Results and Discussion

In our project of Data leakage detection, we presented that whenever someone user share files with others there will be a secret key and if someone want to download or access file, they need a secret key which will be request to sender who share this file, after sharing secret key user can download that file, if someone download that file without asking secret key using guess method that will be notified as leaker.

7. References

[1] N. Sandhya, G. Haricharan Sharma, K. Bhima, "Exerting Modern Techniques for Data Leakage Problems Detect," International Journal of Electronics Communication and Computer Engineering (IJECCE), Vol. 3, Issue(1) NCRTCST, ISSN 2249 – 071X

[2] Sandip A. Kale C1, Prof. S. V. Kulkarni C2, "Data Leakage Detection: A Survey," IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661Vol. 1, Issue 6 (July-Aug 2012), PP 32-35www.iosrjournals.org

[3] P. Saranya, "Online Data Leakage Detection And Analysis," IJART , Vol. 2Issue 2, March 2012

[4] P. Papadimitriou, H. Garcia- Molina, "Data Leakage Detection," technical report, Stanford University, 2008.

[5] Shivappa M. Metagar, Sanjaykumar J. Hamilpure ,B. P. Savukar, "WaterMarking Technique: An Unique Approach For Detecting The Data Leakage,"Volume 2, Issue 5, Sept 2012

[6] Archana Vaidya, Prakash Lahange, Kiran More, Shefali Kachroo & NiveditaPandey, "DATA LEAKAGE DETECTION," Vol. 3, Issue 1, pp. 315-321

[7] Panagiotis Papadimitriou 1, Hector Garcia-Molina 2 Stanford University 353 Serra Street, Stanford, CA 94305, USA P.P (1, 4-5) A Model for Data Leakage Detection

[8]Web-based Data Leakage Prevention Sachiko Yoshihama1, Takuya Mishina1, and Tsutomu Matsumoto2 1 IBM Research - Tokyo, Yamato, Kanagawa, Japan fsachikoy

[9] Joseph A. Rivela Senior Security Consultant P.P (4-6) Data Leakage: Affordable Data Leakage Risk Management

[10] Data Leakage Prevention: A news letter for IT Professionals Issue 5 P.P (1-3)