# TITLE- DATA LEAKAGE DETECTION USING GUILTY MODEL

**GUIDED BY:**
**Prof. Bhulakshmi Bonthu**

**BCI-3001 - Web Security**

**SLOT: B1**
**Fall Semester 2021-2022**

***Rohan Allen (18BCI0247)***
***Rakshith Sachdev (18BCI0109)***
***Rahul Balagopalan (18BCI0157)***

# ABSTRACT

Current statistics from many different security associations and research firms recommend that there has been of late, a really fast growth of data leak in the past 8 years, and as the present world for the most part relies hugely upon exchange of information i.e. transfer of data from one individual or group to another individual or group, which is also known as a distributary system.

Data leakage is a gravely serious security concern for pretty much every organization, as there is almost no organization that does not deal with some kind of data that might be confidential for them or even their customers and clients. The first step to try and solve this problem is to find the initial source of this data leakage. Our project is dealing with shielding the data from being leaked out to outsiders by restricting the agents by using blacklisting so that it cannot be leaked in the first place, thereby not needing any form of mitigation.

Suppose a data distributor has given over some sensitive data to a set of supposedly "trusted" agents (or third parties). Some of the data ends up leaking and found in some unauthorized location (e.g., on the web or some unauthorized 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 are proposing data allocation strategies (across the agents) that improve the probability of identifying these leakages. These methods do not rely wholly 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(s) without much effort. In this day and age where piracy is rampant, it is imperative to protect sensitive data as well as intellectual property from any kind of leak.

# INTRODUCTION

Current statistics from different security association research firms and government organizations recommend that there has been a fast growth of data leak in the past 8 years and as the present world for the most part relies upon exchange of information i.e. transfer of data from one individual to another individual which is also known as distributary system. The data sent from the distributor to the client are confidential so the data is distributed only between the distributor and the trusted third parties.

The data sent by the distributor must be secured, private, confidential and must not be replicated as the data imparted with the trusted third parties are confidential and profoundly significant. In certain events the data distributed by the distributor are duplicated by different agents who cause an enormous harm to the institute and this process of losing the data is known as data leakage. The data leakage must be detected at an early stage in order to prevent the data form being open source. This project deals with shielding the data from being out sourcing by restricting the agents by using blacklisting so that it cannot be leaked.

# PROBLEM STATEMENT

Throughout the course of doing business, once in a while sensitive data must be given over to supposedly trusted third parties. For instance, a medical clinic may give patient records to specialists who will devise new treatments. Similarly, an organization may have partnerships with different organizations that require sharing client data. Another venture may redistribute its data processing, so data must be given to various other organizations. The owner of the data is the distributor and the supposedly trusted third parties are the agents. At that point further data will be given by the distributor to the trusted third party of the enterprise utilizing this application.

We here aim to build an application that will monitor if on the off chance any data has been leaked by the agent of the enterprise. Additionally, here we ensure proper authentication among agents/users accessing the system so that data is accessed by only valid users. It likewise helps in discovering Guilt of Agent from the given set of agents which has leaked the data, who should be blacklisted, using Probability Distribution to find the guilt using the guilt model.

# LITERATURE SURVEY

| PAPER TITLE | METHOD USED | LIMITATION | PROPOSED SYSTEM |
|---|---|---|---|
| Fast Detection of Transformed Data Leaks," in IEEE Transactions on Information Forensics and Security | AlignDLD and Coll inter system<br>(prototype using virtual box | Detects inadvertent leaks and not malicious leaks | Our system deals with both kinds of leaks |
| Data Leakage Detection In cloud using watermarking technique | Hybrid watermarking<br>algorithms | Limited to image data set | Our system deals with any kind of data |
| Data Leakage Detection in cloud computing environment | Bell-Lapadul a Model | Infeasible to extend to web environment where multiple users access data | Our system is feasible for multiple user access |
| Privacy-Preserving Detection<br>of Sensitive Data Exposure | Fuzzy fingerprint technique | False positive and true positive yield the same fingerprints | Our system works on probability to avoid such cases |
| Detection Method on the Privacy Leakage for<br>Composite Services | Detection Method on the Privacy Leakage for<br>Composite Services | Infeasible in case of complex combinations | Our system might get cumbersome, but is feasible for complex combinations |
| ASSESS AGENT GUILT MODEL AND HANDLING DATA ALLOCATION STRATEGIES FOR DATA DISTRIBUTION | Guilty Agent Model | Unusable in case of data theft | Fake objects can be added to improve results |

## EXISTING WORK

There are many reasons why there have been efforts to maintain any leaks in data, and there are similarly many different methods to mitigate in case a leak occurs. There are obviously, in large organisations, systems like a DLP (Data Leak Prevention) system, that does largely what the name suggests, prevent leaks. Companies and organisations employ such systems to be able to contain any breaches or leaks in their system by monitoring all of the traffic that goes from and to the company devices. This could be by reading all emails that go from employees to scan for keywords that may point to some confidential data in the mail, or even decrypting encrypted outgoing transmissions from its pool of keys that are pre-defined and required by employees to use.

But one limitation of DLP comes when the data leaves the hands of the owner organisation or individual, when the DLP can't reach or monitor the data beyond the Home network. This is where methods like DRM (Digital Rights Management) comes in. This is a system that places certain locks on the files, such that only authorized personnel can access it. These use methods such as encryption. It is very common for companies to add DRM to IP software such as video Games or paid software, to protect it against piracy. But this can be hugely cumbersome for the people who use software with DRM, as they can sometimes be very strict, and can be detrimental to the usage of the files in the intended method.

Another very commonly used method to protect data and even find the culprit(s) in case of leaks, is Digital Watermarks. These range from visible to invisible watermarks. Digital Watermarks can be implemented in such a way that a distributor can embed an encrypted signature of the receiver in the data as a watermark before sending it to the receiver. Then, if the receiver illegitimately sends that data to some unauthorised parties, when the leaked data is found, their signature can be recovered to find out the exact source of the leak. But a drawback that is in this method is that the original data has to get modified in order to embed the watermark, whether it is a visible watermark or an invisible watermark.

## CORE MODULES

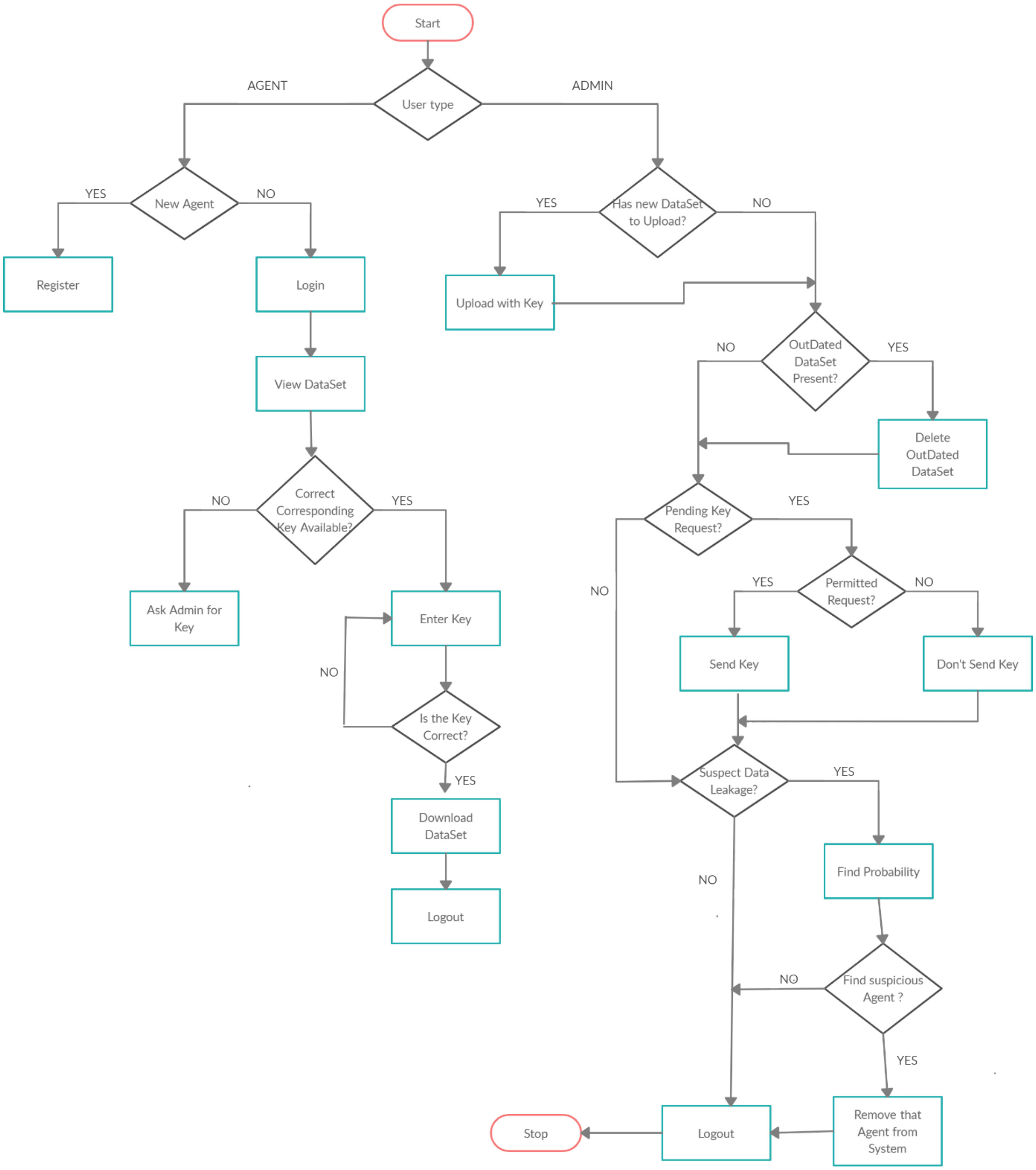
1. Admin Data Control
   This module allows the admin to upload dataset to the database of the system (which can be seen by all users but cannot be accessed without permission) or share any data set to a particular user in private.

2. User Data File Access
   This module allows users to send a request to the admin for a key in order to access the file available in the database of the system. It is only when the proper key is received, the user can access the data file.

3. Probability Of Guilt
   This module analyses which user has the leaked file and sort the list of the probable leakers. Then using the guilt algorithm, the probability calculation is done keeping in mind a cookie jar analogy i.e if we catch Freddie with a single cookie, he can argue that a friend gave him the cookie. But if we catch Freddie with 5 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.

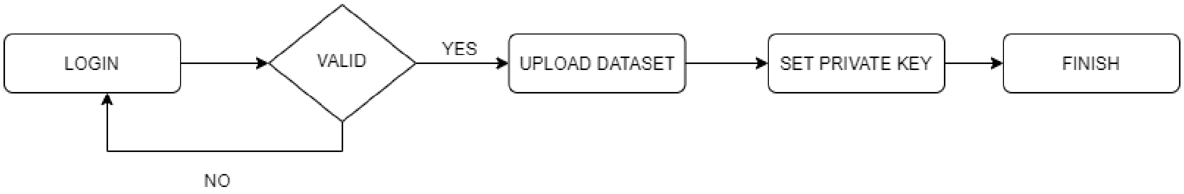
4. Managing the Users
   In this module the admin can make changes to the authority of the users. In other words, he can black list the “known bad” by using the probability of the leaker calculated using the guilt model in order to ensure security of the system.

# ARCHITECTURE DIAGRAM

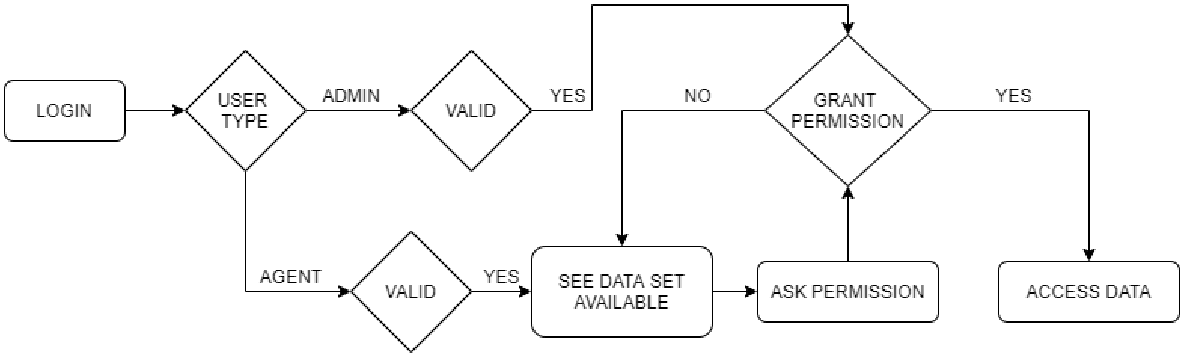


# COMPONENT DIAGRAMS

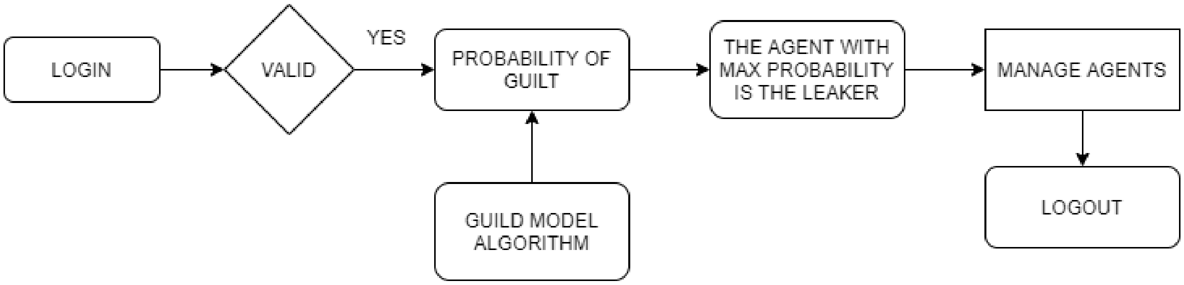
1. Data control



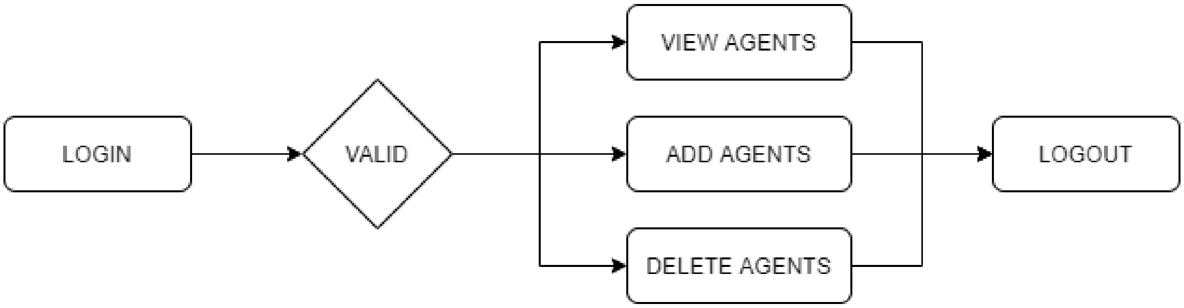
2. File access



3. Probability of guilt



4. Managing agents



## MODULE INTEGRATION

The first module is admin data control which contains the functionality to manage the datasets available for distribution. Admin has the authority to view all the data sets available, remove any data set that he feels are no longer useful and upload new dataset as and when required. Admin controls which agent gets the right to download the datasets, this is done by locking each dataset with their own keys.

To download the dataset the agent must send a request to the admin asking for the key of that dataset. All the requests are visible to the admin and he can send the key to which ever agent that he trusts. Upon receiving the key, the agent is free to download the specific dataset.

In case of data leakage, the guilt module looks into the database to check which agent has the access to what all datasets and uses that information along with the guilty agent model to calculate the probability of guilt of each agent to detect the malicious agent in the system.

Upon find the malicious agent, the admin can use the manage agent module to prohibit them the access to the database.

# IMPLEMENTATION

## PROPOSED ALGORITHM:

To protect the sensitive data, most efficient way is to modify the data and make it “less sensitive”. At times, it is significant not to modify the original distributor’s data. Therefore, here applications where the original sensitive data cannot be perturbed are considered. Generally, leakage detection is taken care of by watermarking, where a unique code is inserted in each distributed copy. On the off chance that that copy is later found in the possession of an unauthorized party, the leaker can be recognized. Watermarks involve some modification of the original data and 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. The model developed is used for assessing the “guilt” of agents. Algorithms are also provided for distributing objects to agents, in a way that improves the chances of identifying a leaker. An option of adding “fake” objects to the distributed set is also considered. Such objects do not compare to genuine entities but seem realistic to the agents. It could be said, the fake objects go about as a sort of watermark for the entire set, without changing any individual members. On the off chance that it turns out an agent was given at least one fake objects that were leaked, then the distributor can be more sure that agent was guilty.

Say the distributor has the set T = {t1, . . . , tm}. The leaked set found out is S. Assumptions made in this implementation are:

1. For all t, t’ ∈ S such that t ≠ t’ the provenance of t is independent of the provenance of t’
2. An object t ∈ S can only be obtained by the target in one of two ways:
   - A single agent Ui leaked t from its own Ri set; or

- The target guessed (or obtained through other means) t without the help of any of the n agents.

Consider that sets T, R's and S are as follows: T = {t1, t2, t3}, R1 = {t1, t2}, R2 = {t1, t3}, S = {t1, t2, t3}. For this situation, every one of the three of the distributor's objects have been leaked and show up in S. Consider how the target may have gotten object t1, which was given to both agents. From Assumption 2, the target either guessed t1 or one of U1 or U2 leaked it. Knowing that the probability of the former event is p, so assuming that probability that each of the two agents leaked t1 is the same cases formed are:

- the target guessed t1 with probability p;
- agent U1 leaked t1 to S with probability $(1 - p)/2$
- agent U2 leaked t1 to S with probability $(1 - p)/2$

Similarly, it is found that agent U1 leaked t2 to S with probability $1 - p$ since he/she is the only agent that has this data object. Given these values, the probability that agent U1 is not guilty can be computed, namely that U1 did not leak either object:

$$P\,r\{G'1|S\} = (1 - (1 - p)/2) \times (1 - (1 - p)) \quad (1)$$

Hence, the probability that U1 is guilty is:

$$P\,r\{G1|S\} = 1 - P\,r\{G'1\} \qquad (2)$$

Consider the set of agents $Vt = \{Ui\ |t \in Ri\}$ that have t in their data sets, now generalizing (1) and (2) :

$$P\,r\{Ui \text{ leaked } t \text{ to } S\} = \{\ 1-p\ /|Vt|\ , \text{ if } Ui \in Vt \text{ and } 0, \text{ otherwise } \} \qquad (3)$$

Given that agent Ui is guilty if he leaks at least one value to S, with Assumption 1 and Equation 3 the probability $P\,r\{Gi\ |S\}$ is computed, that agent Ui is guilty:

$$P\,r\{Gi\ |S\} = 1 - \pi t \in S \cap Ri\ (1 - (1 - p)/\ |Vt|\ )\ (4)$$

## SAMPLE TEST CASES

Table 1 : login

| S.NO | CONDITION /FUNCTIONS | EXPECTED | ACTUAL |
|---|---|---|---|
| 1 | User login :valid credentials | Access granted to the user home page | Access granted to the user home page |
| 2 | User login: invalid credentials | Access denied | Access denied |
| 3 | Admin login: valid credentials | Access granted to the user home page | Access granted to the user home page |
| 4 | Admin login: invalid credentials | Access denied | Access denied |

Table 2: asking permission for access

| S.NO | CONDITION/FUNCTIONS | EXPECTED | ACTUAL |
|---|---|---|---|
| 1 | Private key denied | Access denied to the data set | Access denied to the data set |
| 2 | Private key shared | Access granted to the data set | Access denied to the data set |

Table 3: private key validation

| S.NO | CONDITION/FUNCTIONS | EXPECTED | ACTUAL |
|---|---|---|---|
| 1. | Private key :123<br>Input :124 | Unauthorized access – permission denied | Unauthorized access – permission denied |
| 2 | Private key :123<br>Input:123 | Unauthorized access – | Unauthorized access – |

| | | | |
|---|---|---|---|
| | | permission granted | permission granted |

Table: 4 getting probability

| S.NO | CONDITION/FUNCTIONS | EXPECTED | ACTUAL |
|---|---|---|---|
| 1 | No breach/leakage detected | Probability of guilt :0 (for all) | Probability of guilt :0 (for all) |
| 2 | Leakage/breach detected | Probability of guilt : max (for possible leaker )<br>Min (for innocent agents) | Probability of guilt : max (for possible leaker )<br>Min (for innocent agents) |

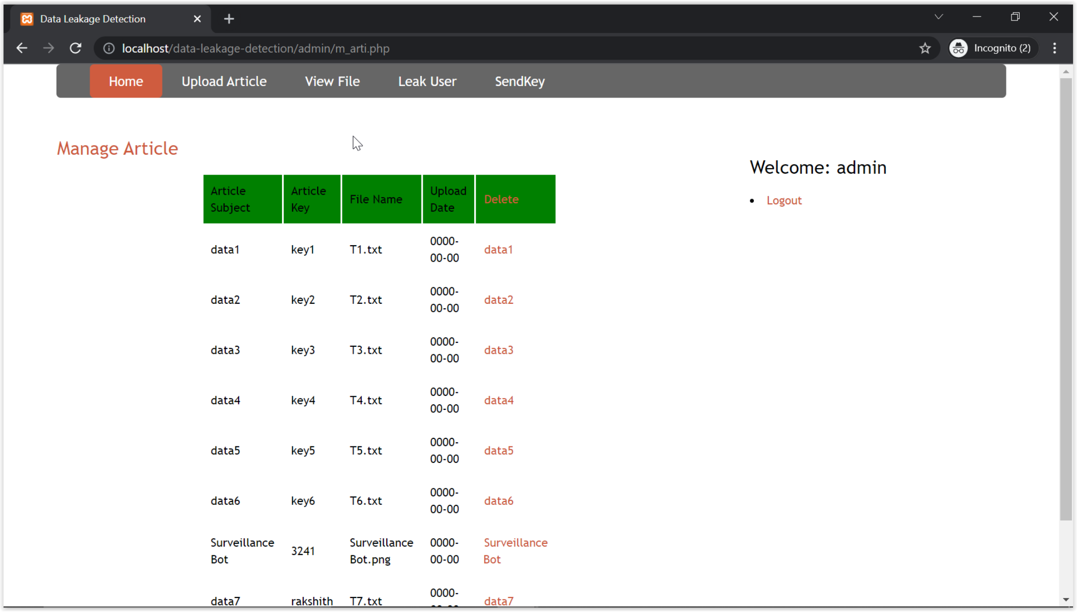
Table5: managing agents

| S.NO | CONDITION/FUNCTIONS | EXPECTED | ACTUAL |
|---|---|---|---|
| 1 | View agents | View list of all registered users | View list of all registered users |
| 2 | Add agents | Add a new user to the system and granting access to basic functionalities | Add a new user to the system and granting access to basic functionalities |
| 3 | Delete agents | Removing a user and denying all access | Removing a user and denying all access |

# OUTPUT SCREENSHOTS

Module 1: Data control by the admin

(A) Management of all the dataset present on the database.



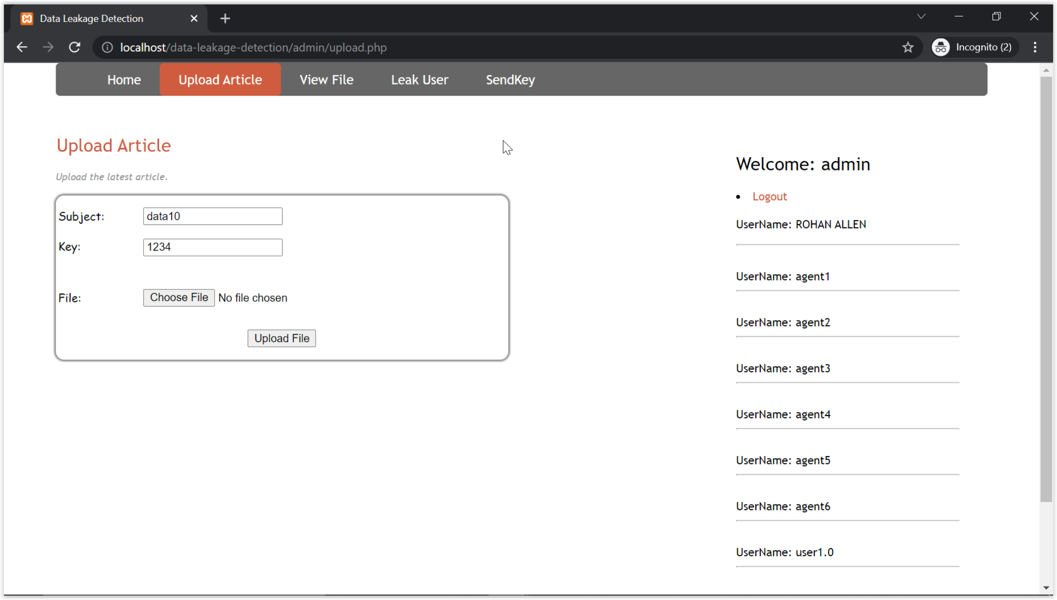
Manage Article

| Article Subject | Article Key | File Name | Upload Date | Delete |
|---|---|---|---|---|
| data1 | key1 | T1.txt | 0000-00-00 | data1 |
| data2 | key2 | T2.txt | 0000-00-00 | data2 |
| data3 | key3 | T3.txt | 0000-00-00 | data3 |
| data4 | key4 | T4.txt | 0000-00-00 | data4 |
| data5 | key5 | T5.txt | 0000-00-00 | data5 |
| data6 | key6 | T6.txt | 0000-00-00 | data6 |
| Surveillance Bot | 3241 | Surveillance Bot.png | 0000-00-00 | Surveillance Bot |
| data7 | rakshith | T7.txt | 0000- | data7 |

Welcome: admin

- Logout

(B) Uploading new dataset to the database



Upload Article

*Upload the latest article.*

Subject: data10

Key: 1234

File: No file chosen

Welcome: admin

- Logout

UserName: ROHAN ALLEN

UserName: agent1

UserName: agent2

UserName: agent3

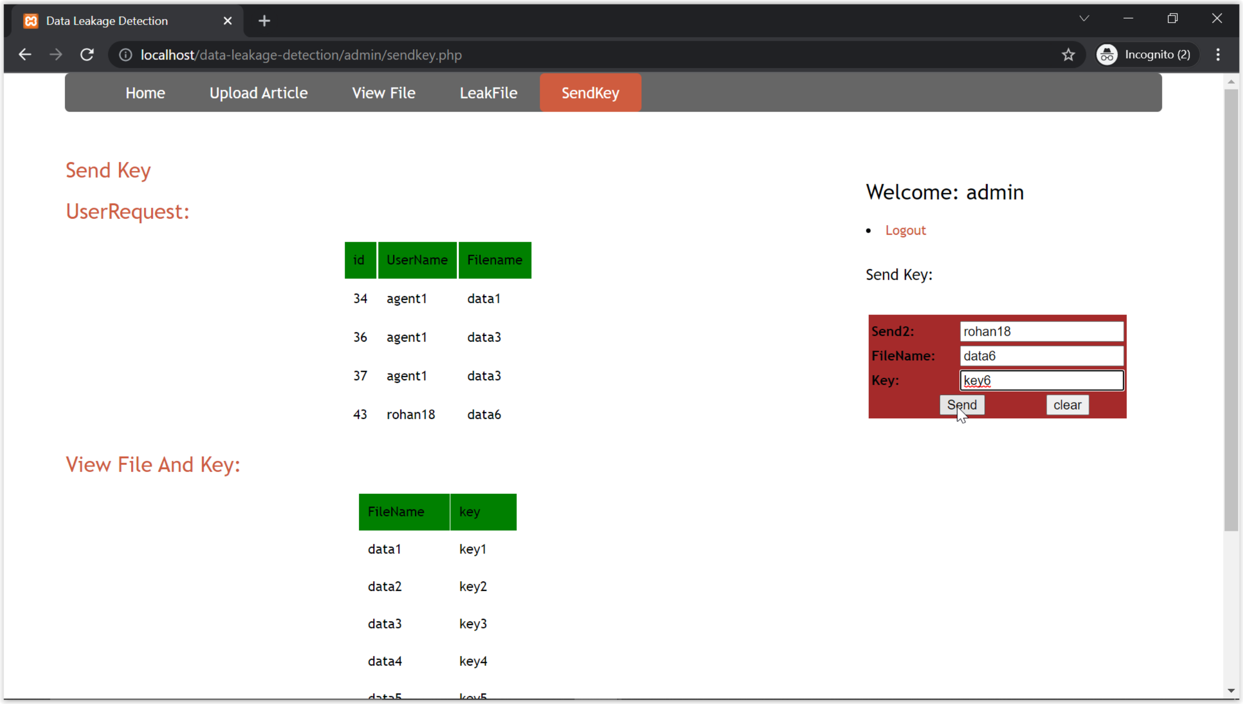
UserName: agent4

UserName: agent5

UserName: agent6

UserName: user1.0

(C) Sending the keys to as per requirement and trust between the admin and the agent



Send Key

UserRequest:

| id | UserName | Filename |
|---|---|---|
| 34 | agent1 | data1 |
| 36 | agent1 | data3 |
| 37 | agent1 | data3 |
| 43 | rohan18 | data6 |

View File And Key:

| FileName | key |
|---|---|
| data1 | key1 |
| data2 | key2 |
| data3 | key3 |
| data4 | key4 |
| data5 | key5 |

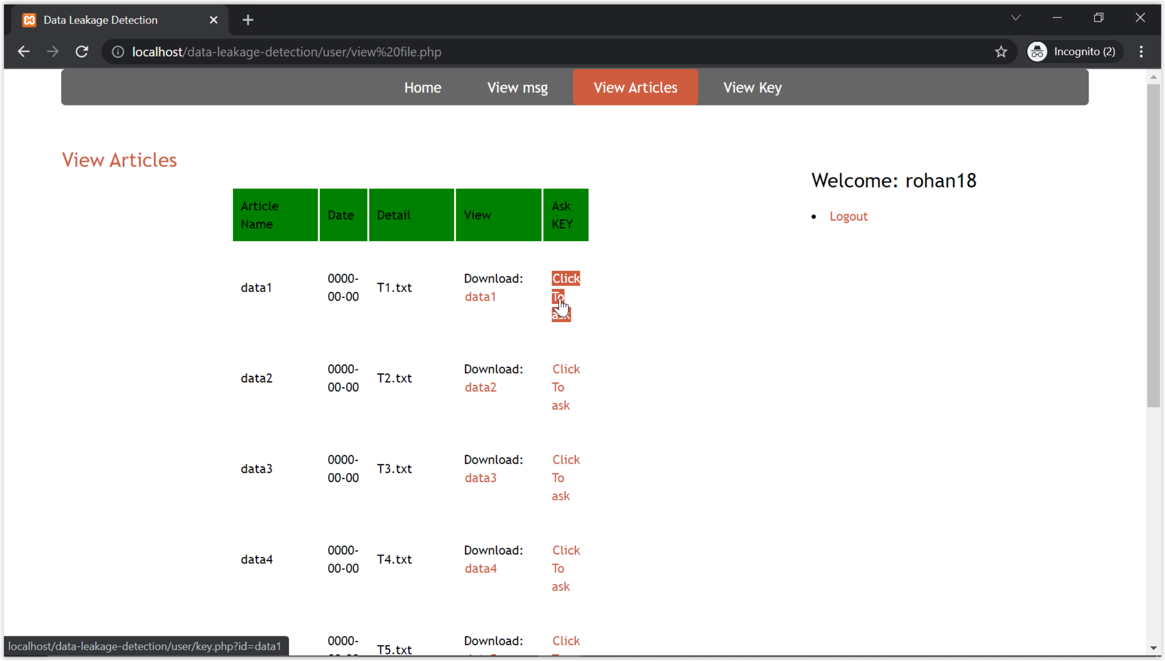
Welcome: admin

Send Key:

| | |
|---|---|
| Send2: | rohan18 |
| FileName: | data6 |
| Key: | key6 |

Module 2: Accessing of files by the user

(A) Asking for key for the required dataset to the admin

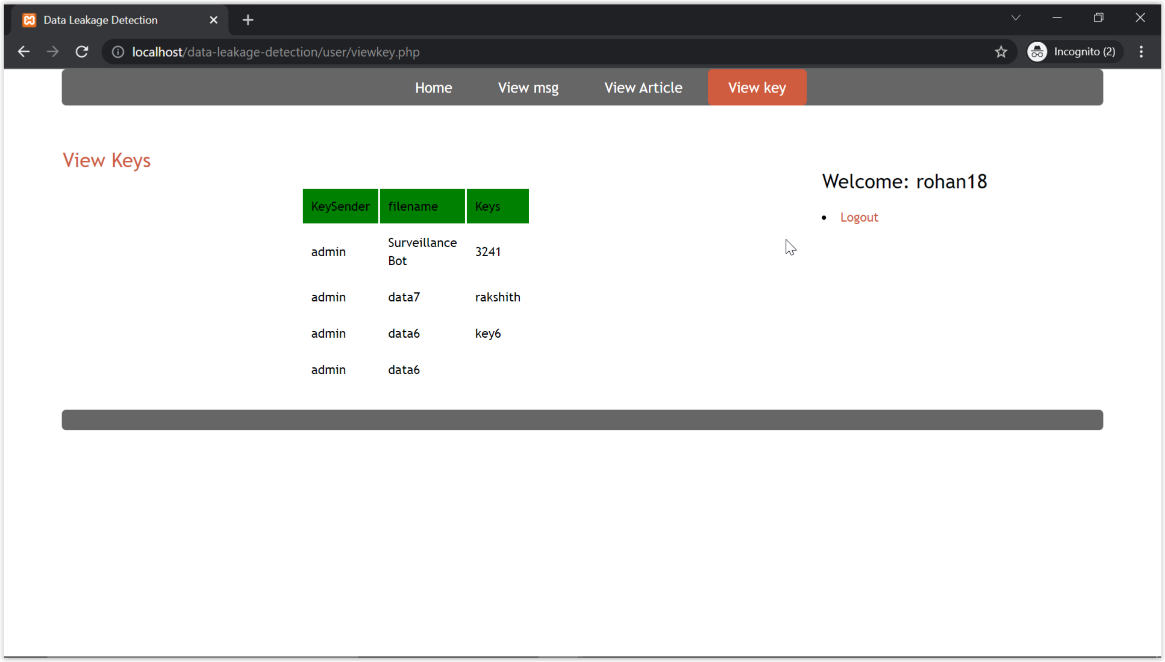

Data Leakage Detection
localhost/data-leakage-detection/user/view%20file.php
Incognito (2)
Home
View msg
View Articles
View Key
Logout
localhost/data-leakage-detection/user/key.php?id=data1

View Articles

Welcome: rohan18

| Article Name | Date | Detail | View | Ask KEY |
|---|---|---|---|---|
| data1 | 0000-00-00 | T1.txt | Download: data1 | Click To ask |
| data2 | 0000-00-00 | T2.txt | Download: data2 | Click To ask |
| data3 | 0000-00-00 | T3.txt | Download: data3 | Click To ask |
| data4 | 0000-00-00 | T4.txt | Download: data4 | Click To ask |
|  | 0000- | T5.txt | Download: | Click |

(B) Accessing the key send by the admin

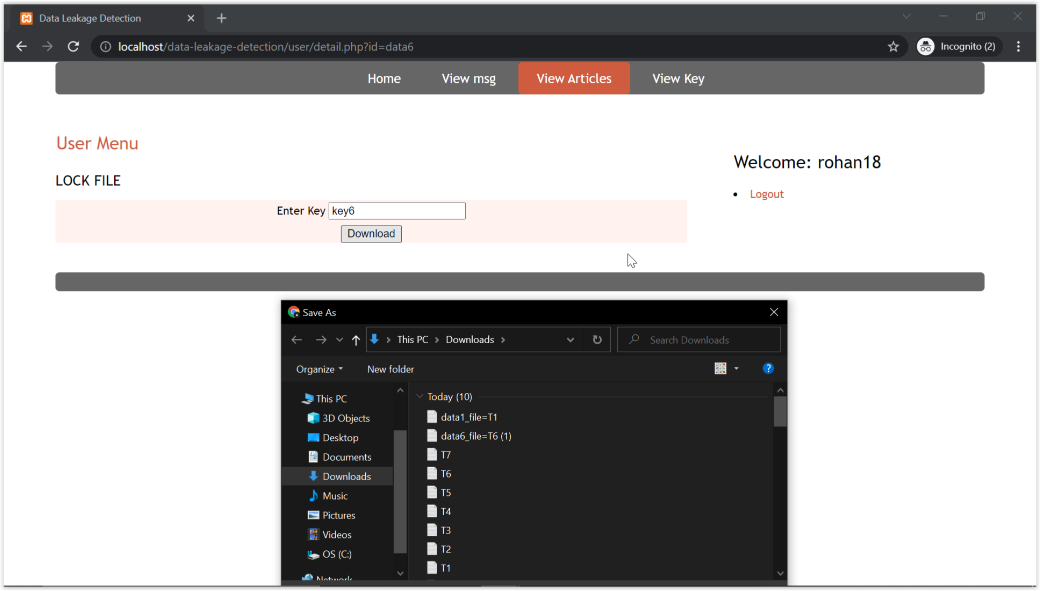


View Keys

Welcome: rohan18

| KeySender | filename | Keys |
|---|---|---|
| admin | Surveillance Bot | 3241 |
| admin | data7 | rakshith |
| admin | data6 | key6 |
| admin | data6 |  |

(C) Downloading the dataset



localhost/data-leakage-detection/user/detail.php?id=data6

User Menu

LOCK FILE

Enter Key: key6

Welcome: rohan18

This PC > Downloads

Today (10)
- data1_file=T1
- data6_file=T6 (1)
- T7
- T6
- T5
- T4
- T3
- T2
- T1

Module 3: Finding the probability of guilt of each agent when a particular dataset has been found with unauthorized entity

Let all the agents have the following datasets:

Agent1={t2,t4,t6,t8,t10,t12}

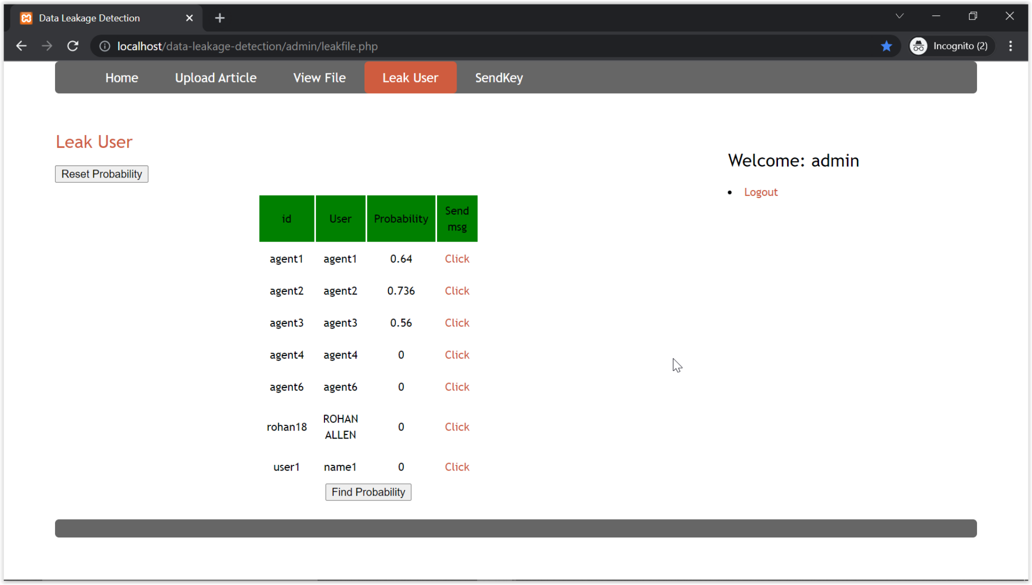
Agent2={t6,t7,t8,t9,t10}

Agent3={t1,t7,t9,t4,t5}

Agent4={t1,t3,t5,t7,t9,t11}

And the leaked dataset be S={t1,t6,t9,t10}

Then the guilt probability would be



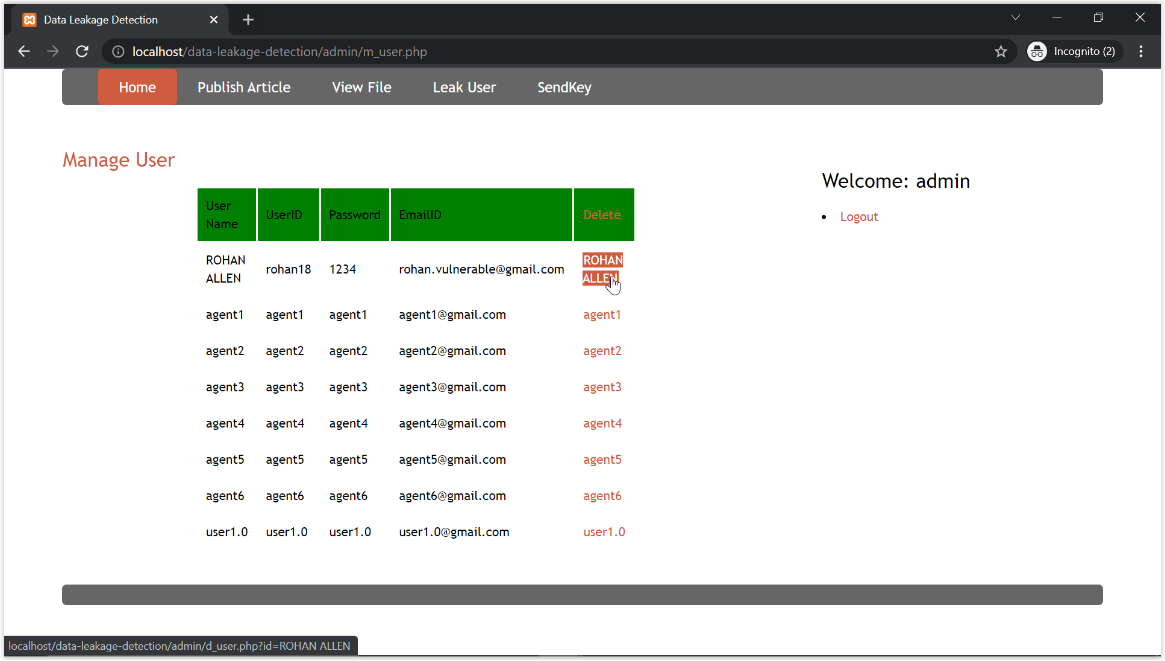
localhost/data-leakage-detection/admin/leakfile.php

Leak User

Welcome: admin

| id | User | Probability | Send msg |
|---|---|---|---|
| agent1 | agent1 | 0.64 | Click |
| agent2 | agent2 | 0.736 | Click |
| agent3 | agent3 | 0.56 | Click |
| agent4 | agent4 | 0 | Click |
| agent6 | agent6 | 0 | Click |
| rohan18 | ROHAN ALLEN | 0 | Click |
| user1 | name1 | 0 | Click |

Module 4: Managing of user which allows admin to blacklist any agent who is not trustworthy


Data Leakage Detection
localhost/data-leakage-detection/admin/m_user.php
Incognito (2)
Home
Publish Article
View File
Leak User
SendKey
Logout
localhost/data-leakage-detection/admin/d_user.php?id=ROHAN ALLEN

Manage User

Welcome: admin

| User Name | UserID | Password | EmailID | Delete |
|---|---|---|---|---|
| ROHAN ALLEN | rohan18 | 1234 | rohan.vulnerable@gmail.com | ROHAN ALLEN |
| agent1 | agent1 | agent1 | agent1@gmail.com | agent1 |
| agent2 | agent2 | agent2 | agent2@gmail.com | agent2 |
| agent3 | agent3 | agent3 | agent3@gmail.com | agent3 |
| agent4 | agent4 | agent4 | agent4@gmail.com | agent4 |
| agent5 | agent5 | agent5 | agent5@gmail.com | agent5 |
| agent6 | agent6 | agent6 | agent6@gmail.com | agent6 |
| user1.0 | user1.0 | user1.0 | user1.0@gmail.com | user1.0 |

## RESULT

We here are hence, successful in building an application that will monitor if any data has been leaked by the agent of the enterprise. It likewise helps in discovering possible Guilt of an Agent from the given set of agents which has leaked the data using Probability Distribution. This result is achieved without any tampering of the original data, which is similar to what happens in methods like digital watermarking.

## CONCLUSION AND FUTURE WORK

In future, along with finding the probability of guilt model, concept of a fake agent /data set can be implemented by adding fake data into the sensitive data set ,which will act as a watermark without changing the data itself .This technique will further ease the process of detecting leakage or a leaker .

## References:

[1] X. Shu, et al., “Fast Detection of Transformed Data Leaks,” in IEEE Transactions on
Information Forensics and Security, vol. 11, no. 3, pp. 528-542, 2016.

[2] X. Shu, et al., “Privacy-preserving detection of sensitive data exposure,” IEEE Trans.
Inf. Forensics Security, vol. 10, no. 5, pp. 1092-1103, 2015.

[3] H. A. Kholidy, et al., “DDSGA: A data-driven semi-global alignment approach for detecting masquerade attacks,” IEEE Trans. Dependable Secure Comput., vol. 12, no. 2, pp. 164-178, 2015.

[4] Panagiotis Papadimitriou, Student Member, IEEE, and Hector Garcia-Molina, Member,
IEEE “Data Leakage Detection” IEEE Transactions on Knowledge and Data Engineering,
Vol. 23, NO. 1, JANUARY 2011

[5] Amir Harel, Asaf Shabtai, LiorRokach, and Yuval Elovici “M-Score: A Misuseability Weight Measure” IEEE Transactions ON Dependable And Secure Computing, Vol.9, NO. 3, MAY/JUNE 2012

[6] V. Dhanalakshmi and R. Shagana, "Assess agent guilt model and handling data allocation strategies for data distribution," 2013 International Conference on Optical Imaging Sensor and Security (ICOSS), Coimbatore, 2013, pp. 1-5, doi: 10.1109/ICOISS.2013.6678421.