DATA LEAKAGE DETECTION

ABSTRACT

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 cases we can also inject “realistic but fake” data records to further improve our chances of detecting leakage and identifying the guilty party.

OBJECTIVE

- 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 cases we can also inject “realistic but fake” data records to further improve our chances of detecting leakage and identifying the guilty party.

- 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.

EXISTING SYSTEM

- 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.

- However, in some cases it is important not to alter the original distributor’s data.

- 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.

PROPOSED SYSTEM

- After giving a set of objects to agents, the distributor discovers some of those same objects in an unauthorized place.

- 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.

- 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 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.

- 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.

Hardware Requirements

- **SYSTEM : Pentium IV 2.4 GHz**
- **HARD DISK : 40 GB**
- **FLOPPY DRIVE : 1.44 MB**
- **MONITOR : 15 VGA colour**
- **MOUSE : Logitech.**
- **RAM : 256 MB**
- **KEYBOARD : 110 keys enhanced.**

Software Requirements

- **Operating system :- Windows XP Professional**
- **Front End :- Microsoft Visual Studio .Net 2005**
- **Coding Language :- C#**
- **Database :- SQL SERVER 2000**

MODULE DESCRIPTION:

1) Login / Registration:

This is a module mainly designed to provide the authority to a user in order to access the other modules of the project. Here a user can have the accessibility authority after the registration.

2) DATA TRANSFER:

This module is mainly designed to transfer data from distributor to agents. The same module can also be used for illegal data transfer from authorized to agents to other agents

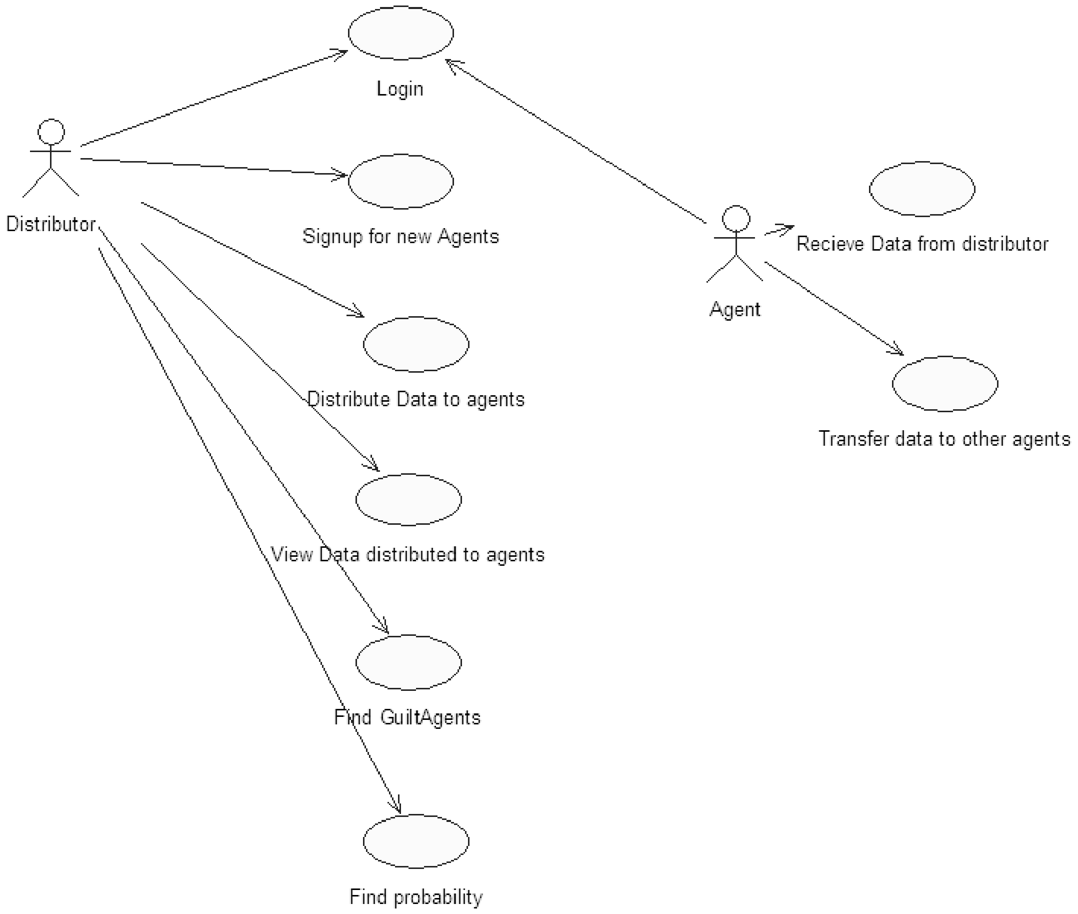
3) GUILT MODEL ANALYSIS:

This module is designed using the agent – guilt model. Here a count value(also called as fake objects) are incremented for any transfer of data occurrence when agent transfers data. Fake objects are stored in database.

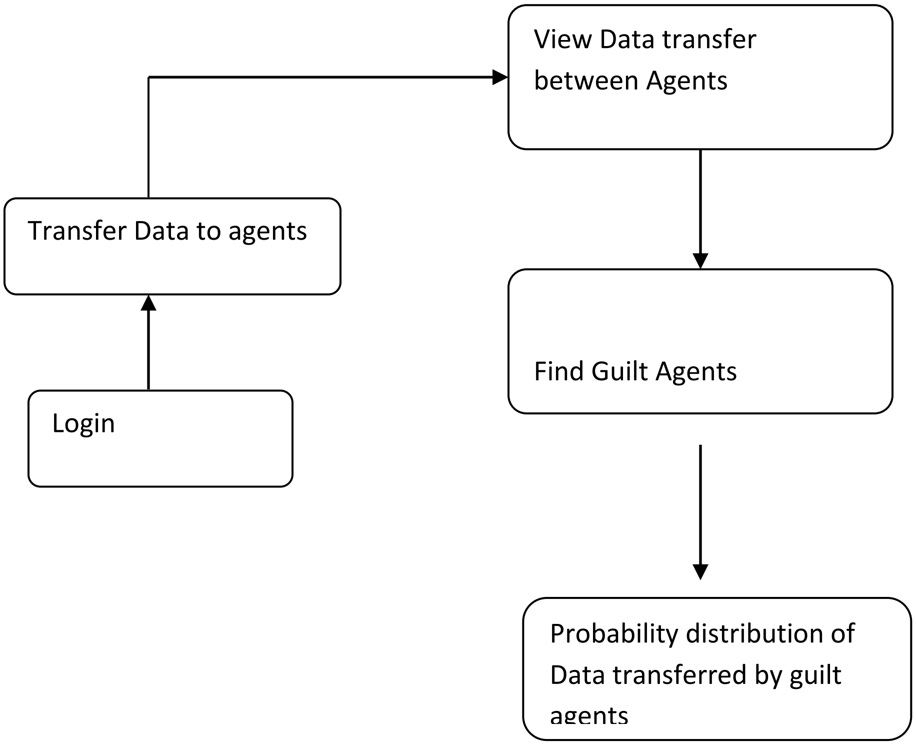
4)AGENT-GUILT MODEL:

This module is mainly designed for determining fake agents. This module uses fake objects (which is stored in database from guilt model module) and determines the guilt agent along with the probability. A graph is used to plot the probability distribution of data which is leaked by fake agents

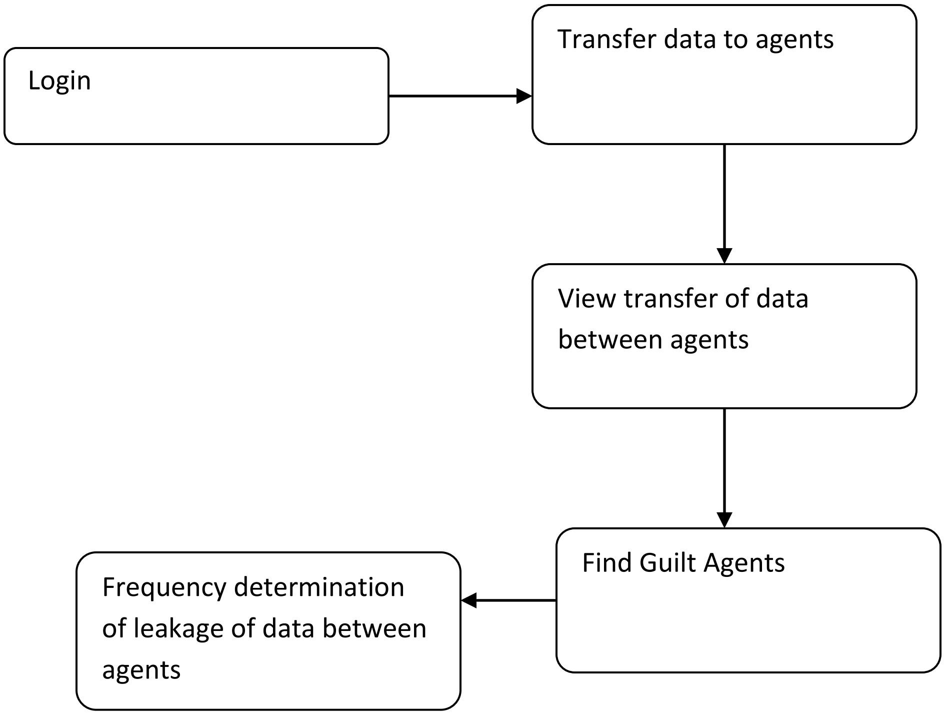
USE CASE DIAGRAM:


Login
Distributor
Signup for new Agents
Agent
Recieve Data from distributor
Distribute Data to agents
Transfer data to other agents
View Data distributed to agents
Find GuiltAgents
Find probability

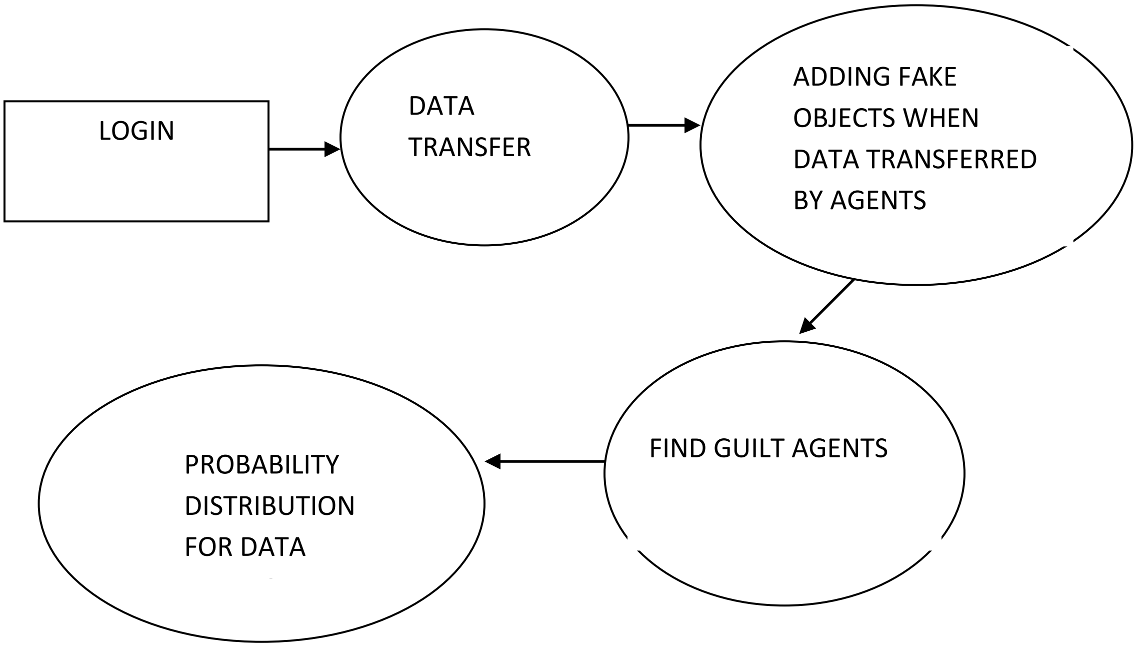
MODULE DIAGRAM:


View Data transfer between Agents
Transfer Data to agents
Find Guilt Agents
Login
Probability distribution of Data transferred by guilt agents

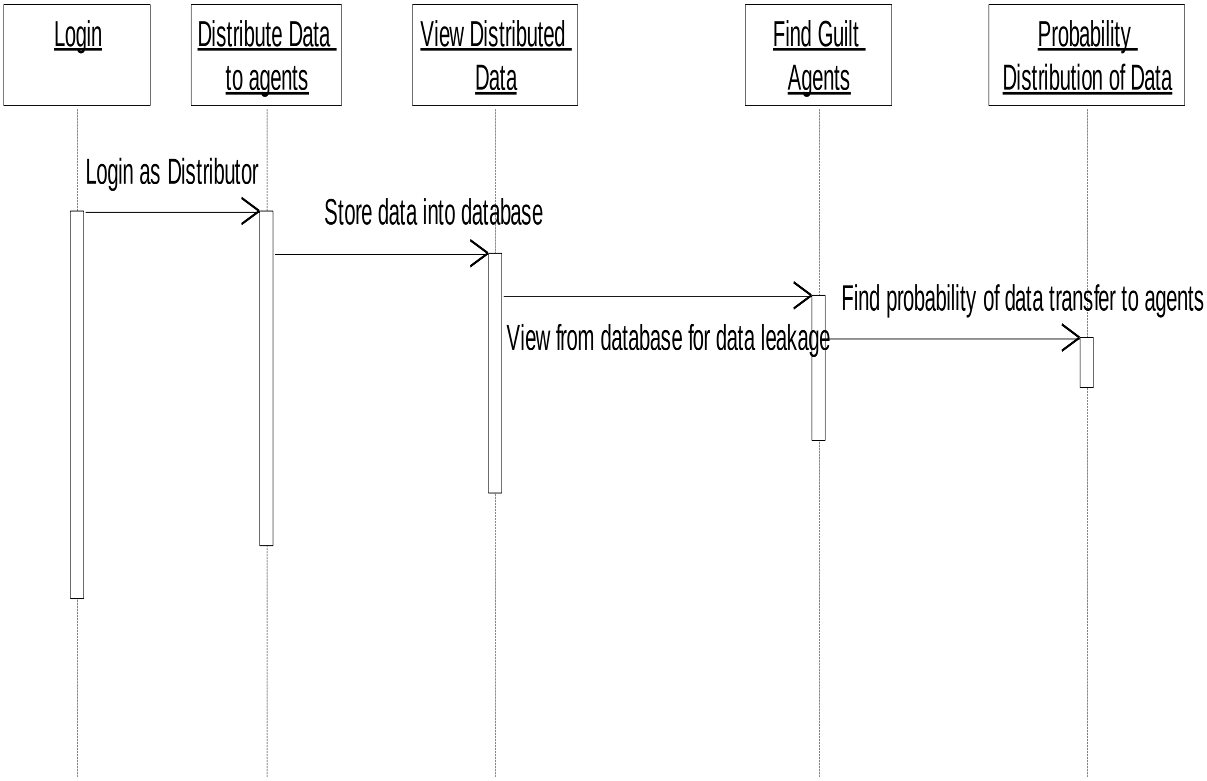
Object Diagram:


Login
Transfer data to agents
View transfer of data between agents
Find Guilt Agents
Frequency determination of leakage of data between agents

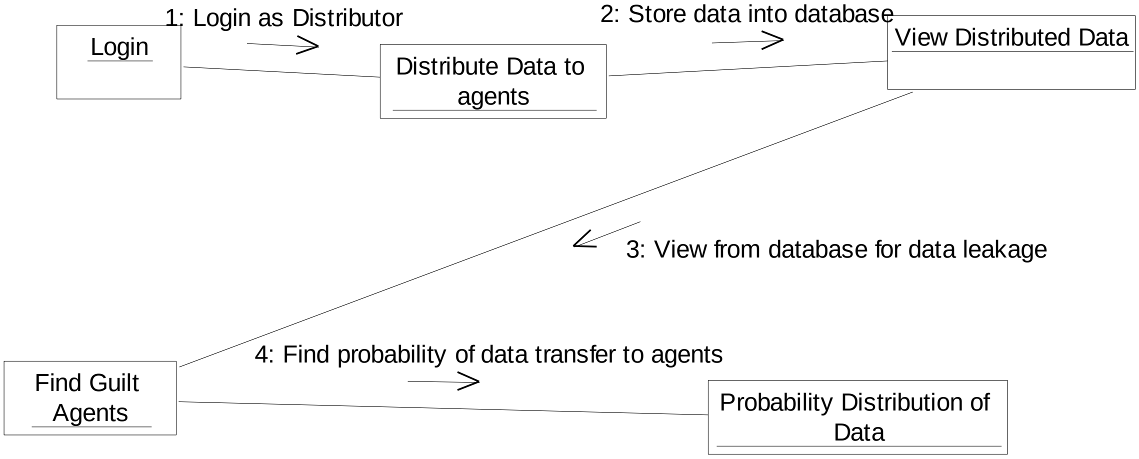
Project Flow Diagram:


LOGIN
DATA TRANSFER
ADDING FAKE OBJECTS WHEN DATA TRANSFERRED BY AGENTS
FIND GUILT AGENTS
PROBABILITY DISTRIBUTION FOR DATA

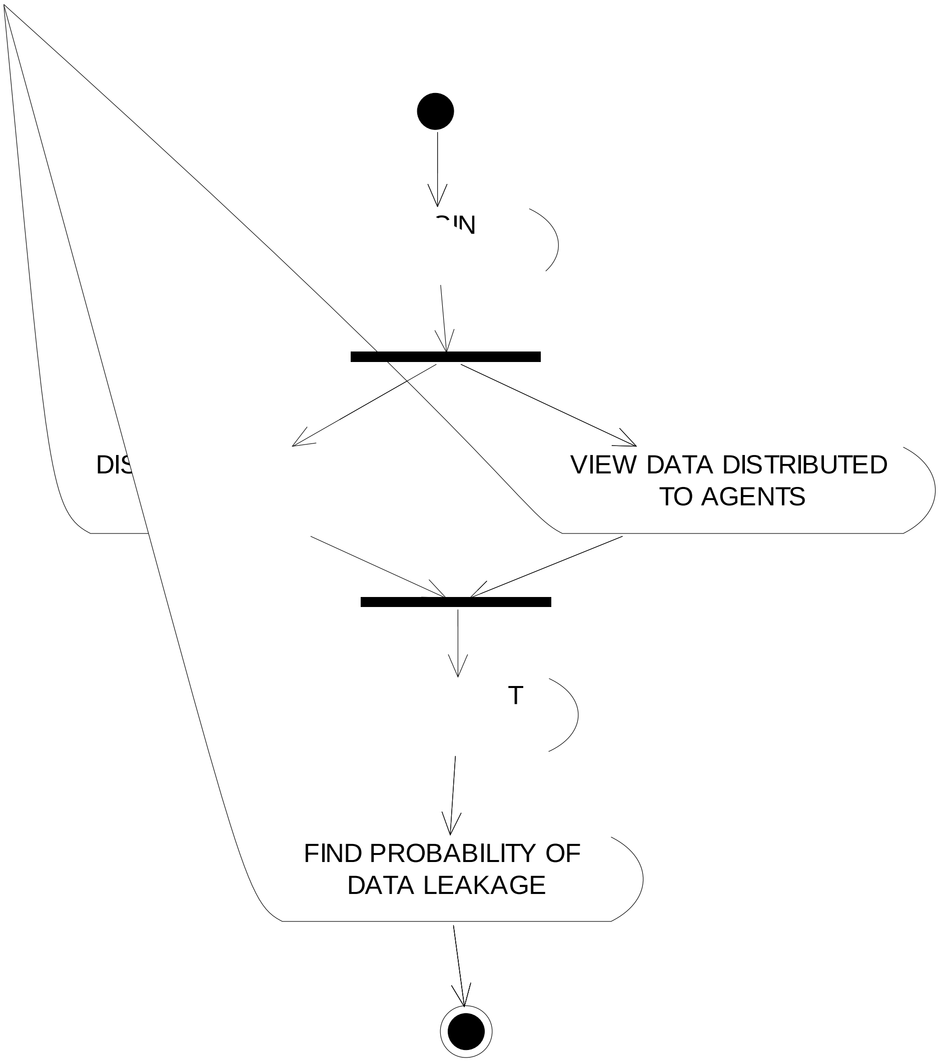
SEQUENCE DIAGRAM:


Login
Distribute Data to agents
View Distributed Data
Find Guilt Agents
Probability Distribution of Data
Login as Distributor
Store data into database
View from database for data leakage
Find probability of data transfer to agents

COLLABORATION DIAGRAM:

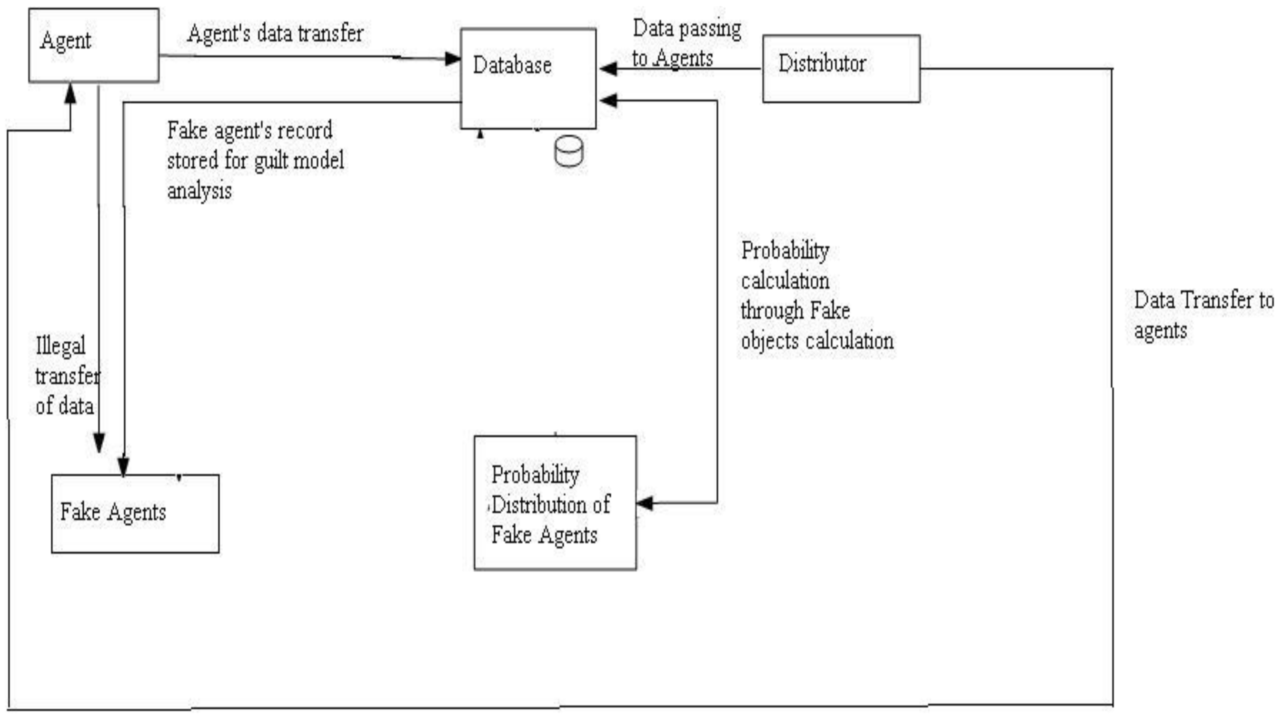

1: Login as Distributor
2: Store data into database
Login
Distribute Data to agents
View Distributed Data
3: View from database for data leakage
4: Find probability of data transfer to agents
Find Guilt Agents
Probability Distribution of Data

ACTIVITY DIAGRAM:

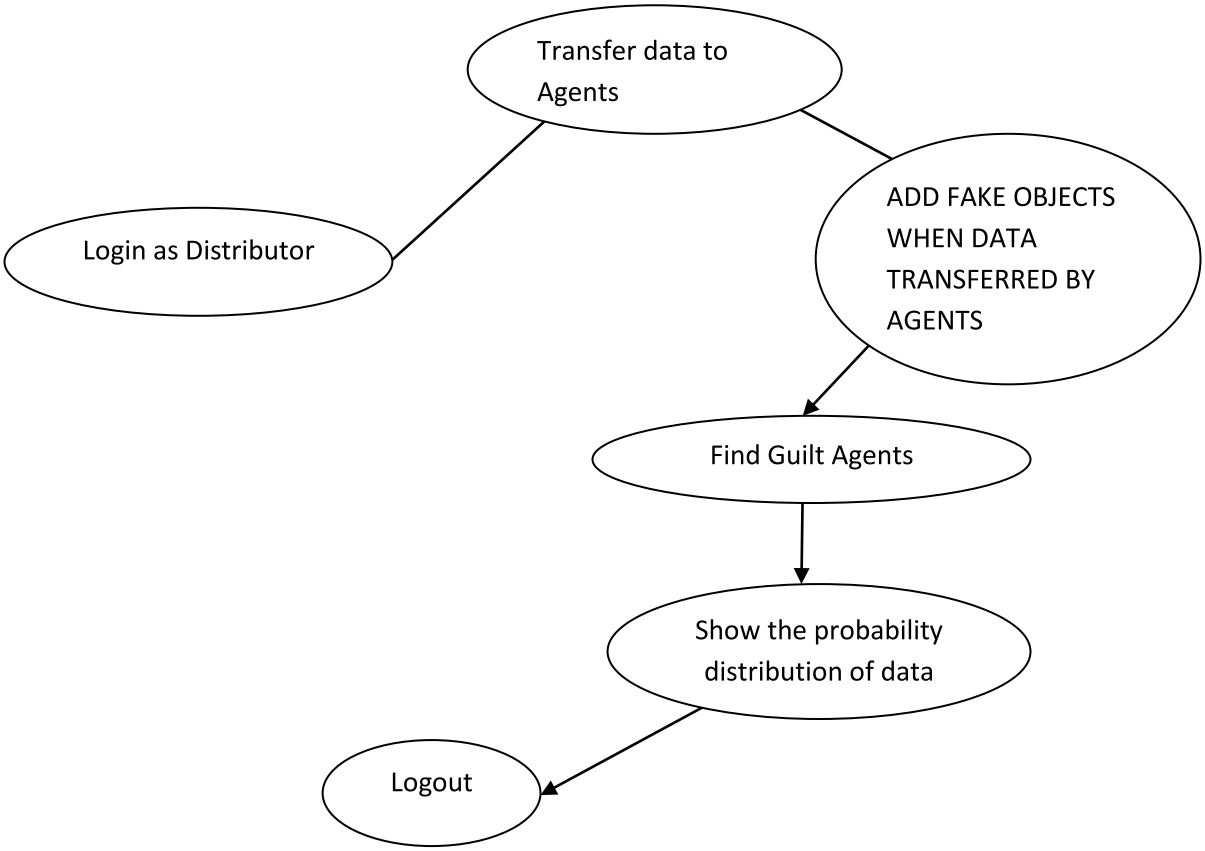


GIN
DIS
VIEW DATA DISTRIBUTED TO AGENTS
T
FIND PROBABILITY OF DATA LEAKAGE

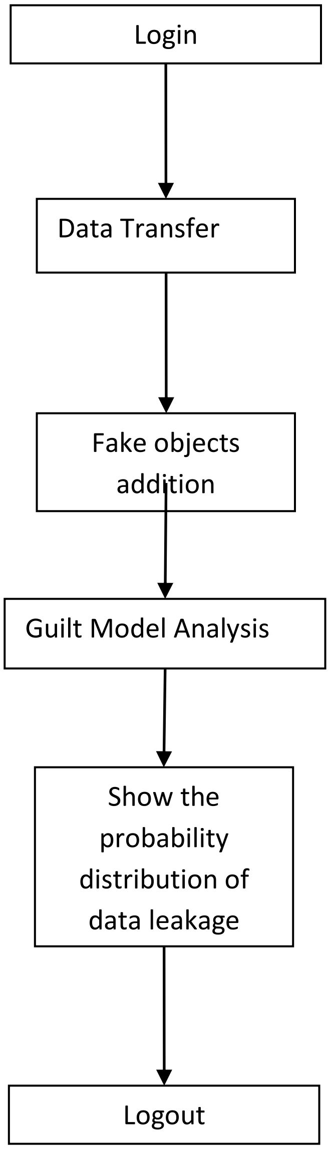
ARCHITECTURE DIAGRAM:



E-R DIAGRAM:


Transfer data to Agents
Login as Distributor
ADD FAKE OBJECTS WHEN DATA TRANSFERRED BY AGENTS
Find Guilt Agents
Show the probability distribution of data
Logout

DATA FLOW DIAGRAM:



CONCLUSION

The likelihood that an agent is responsible for a leak is assessed, 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. 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.

REFERENCES

[1] R. Agrawal and J. Kiernan. Watermarking relational databases. In VLDB ’02: Proceedings of the 28th international conference on Very Large Data Bases, pages 155–166. VLDB Endowment, 2002.

[2] P. Bonatti, S. D. C. di Vimercati, and P. Samarati. An algebra for composing access control policies. ACM Trans. Inf. Syst. Secur., 5(1):1–35, 2002.

[3] P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. In J. V. den Bussche and V. Vianu, editors, Database Theory - ICDT 2001, 8th International Conference, London, UK, January 4-6, 2001, Proceedings, volume 1973 of Lecture Notes in Computer Science, pages 316–330. Springer, 2001.

[4] P. Buneman and W.-C. Tan. Provenance in databases. In SIGMOD ’07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pages 1171–1173, New York, NY, USA, 2007. ACM.

[5] Y. Cui and J. Widom. Lineage tracing for general data warehouse transformations. In The VLDB Journal, pages 471–480, 2001.

KEY TERMS

Data Leakage:

A data breach is the unintentional release of secure information to an untrusted environment.

Data Privacy:

Information privacy, or data privacy is the relationship between collection and dissemination of data, technology, the public expectation of privacy, and the legal and political issues surrounding them.

Privacy concerns exist wherever personally identifiable information is collected and stored - in digital form or otherwise. Improper or non-existent disclosure control can be the root cause for privacy issues.

Fake records:

Records which are false or containing misleading appearance.

Unobstrusive Techniques:

Unobtrusive technique is a technique of data collection. They describe methodologies which do not involve direct elicitation of data from the research subjects. The unobtrusive approach often seeks unusual data sources.

WHY USE DATAMINING

Data mining is the process of extracting patterns from data. Data mining is becoming an increasingly important tool to transform the data into information. It is commonly used in a wide range of profiling practices, such as marketing, surveillance, fraud detection and scientific discovery.

Data mining can be used to uncover patterns in data but is often carried out only on samples of data. The mining process will be ineffective if the samples are not a good representation of the larger body of data. Data mining cannot discover patterns that may be present in the larger body of data if those patterns are not present in the sample being "mined". Inability to find patterns may become a cause for some disputes between customers and service providers. Therefore data mining is not foolproof but may be useful if sufficiently representative data samples are collected. The discovery of a particular pattern in a particular set of data does not necessarily mean that a pattern is found elsewhere in the larger data from which that sample was drawn. An important part of the process is the verification and validation of patterns on other samples of data