**Data Leakage Detection**

In today’s age, most of the companies (medium or large) outsource their business or human resources operations to a third party, also called as Business Process Outsourcing (BPO). It is very useful as it saves time and operations cost and makes them to focus on their core competencies. But, this gives rise to security risks as the third party may require access to sensitive data such as employee details, customer details, intellectual property details and other confidential information. But critical problem in BPO is that the service provider may not be fully trusted or may not be securely administered. Business agreements for BPO try to regulate how the data will be handled by service providers, but it is almost impossible to truly enforce or verify such policies across different administrative domains. Due to their digital nature, relational databases (that most of the organization use) are easy to duplicate and in many cases a service provider may have financial interests to redistribute commercially valuable data or may simply fail to handle it properly. Hence, we need powerful techniques that can detect such dishonest people, so that the losses that may arise from them may be recovered from them and no further business would be done with them that may prevent future losses. After having studied the problem, let us find the possible techniques for data leakage detection:

1. **Using Watermarks**

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 along with addition of some overheads. So, they are generally used in audio, video and images. 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.

1. **Adding ‘Fake’ Objects**

Another technique for leakage detection is adding fake data while transferring data to the agents that helps detecting ‘guilt’ agent by checking if the leaked data has the fake object along with the original data.

Let us study the ‘fake’ object model in detail:

If there is a distributor that owns a set T = {t1,…,tm} of valuable data objects. The distributor wants to share some of the objects with a set of agents U1, U2, …Un, but does not wish the objects be leaked to other third parties. The objects in T could be of any type and size, e.g., they could be tuples in a relation, or relations in a database. An agent Ui receives a subset of T, which can either be a sample request or an explicit request. This involves complex mathematical operations that help in analysis of leaked data to fine the guilt agent.

**MODULES**:

1. Data Allocation Module:

The main focus of this technique is the data allocation problem as how can the distributor “intelligently” give data to agents in order to improve the chances of detecting a guilty agent.

2. Fake Object Module:

Fake objects are objects generated by the distributor in order to increase the chances of detecting agents that leak data. The distributor may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. Our use of fake objects is inspired by the use of “trace” records in mailing lists.

3. Optimization Module:

The Optimization Module is the distributor’s data allocation to agents has one constraint and one objective. The distributor’s constraint is to satisfy agents’ requests, by providing them with the number of objects they request or with all available objects that satisfy their conditions. His objective is to be able to detect an agent who leaks any portion of his data.

4. Data Distributor

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.

In doing a business there would be no need to hand over 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. 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. The second model is relatively simple, but we believe it captures the essential trade-offs. The algorithms that are presented implement a variety of data distribution strategies that can improve the distributor’s chances of identifying a leaker. It has shown that distributing objects judiciously can make a significant difference in identifying guilty agents, especially in cases where there is large overlap in data that agents must receive.